



Guida per gli sviluppatori

Amazon GameLift



Amazon GameLift: Guida per gli sviluppatori

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Che cos'è Amazon GameLift?	1
Usi di Amazon GameLift	1
Inizia a usare le GameLift soluzioni Amazon	1
GameLiftHosting Amazon per server personalizzati	2
GameLiftHosting Amazon con server in tempo reale	2
Amazon GameLift FleetIQ per l'hosting su Amazon EC2	3
Amazon GameLift FlexMatch per il matchmaking	3
Hosting GameLift Anywhere hardware Amazon	4
Accedere ad Amazon GameLift	4
Prezzi per Amazon GameLift	5
Come GameLift funziona Amazon	5
Componenti chiave	5
Hosting di server di gioco	6
Sessioni di gioco in corso	7
Dimensionamento della capacità del parco istanze	7
Monitoraggio di Amazon GameLift	8
Utilizzo di altre AWS risorse	8
Come i giocatori si connettono ai giochi	9
Architettura di gioco con Amazon gestito GameLift	10
Configurazione	13
Configurazione di un account	13
Registrati per un Account AWS	14
Crea un utente con accesso amministrativo	14
Gestisci le autorizzazioni degli utenti per Amazon GameLift	15
Configura l'accesso programmatico per gli utenti	16
Configura l'accesso programmatico per il tuo gioco	18
Esempi di autorizzazioni IAM	19
Imposta un ruolo di servizio IAM	23
Supporto allo sviluppo	26
Per server di gioco personalizzati	26
Per servizi client personalizzati	28
Per server in tempo reale	28
Gestisci i costi di hosting dei giochi	29
Crea avvisi di fatturazione per monitorare l'utilizzo	29

Tieni traccia dei costi per GameLift flotta Amazon	30
Imposta a zero la capacità della flotta inutilizzata	30
Sedi GameLift di hosting Amazon	30
GameLift Hosting su Amazon	31
Zone locali	32
Amazon GameLift Anywhere	34
Amazon GameLift FlexMatch	34
Amazon GameLift in Cina	34
Nozioni di base	35
Esempio di server di gioco personalizzato	35
Esempio di gioco Realtime Servers	35
Roadmap per l'hosting gestito	37
Scegli un'opzione di hosting	37
Prepara il tuo gioco	39
Prepara il tuo server di gioco personalizzato	39
Prepara il tuo server in tempo reale	40
Testa la tua integrazione	40
Pianifica e distribuisce le tue risorse	41
Distribuisce le tue risorse	42
Progetta il tuo servizio di backend	42
Autenticazione dei giocatori	43
Backend senza server	43
WebSocketbackend basato	45
Configurazione delle metriche e della registrazione	47
Lancia liste di controllo	48
Onboarding	48
Test	49
Avvia	50
Dopo il lancio	50
Preparazione di giochi per Amazon GameLift	51
Integra i giochi con server di gioco personalizzati	51
GameLiftInterazioni con Amazon	52
Integra un server di gioco	56
Integra un client di gioco	67
Motori di gioco e Amazon GameLift	74
Testa la tua integrazione (server SDK 5)	100

Testa la tua integrazione (server SDK 4)	108
Integrazione di giochi con server in tempo reale	116
Cosa sono i server in tempo reale?	116
Gestione delle sessioni di gioco	117
Interazione client-server	117
Personalizzazione di un server	118
Implementazione e aggiornamento	119
Integrazione di un client di gioco	119
Personalizzazione di uno script in tempo reale	125
Integrazione dei giochi con il plug-in per Unity	131
Guida ai plugin per Unity (server SDK 5.x)	131
Guida ai plugin per Unity (server SDK 4.x)	150
Integrazione dei giochi con il plug-in per Unreal	177
Informazioni sul plugin	178
Workflow del plugin	179
Installa il plugin per Unreal	179
Configura un profilo AWS utente	183
Configura il gioco con Anywhere	185
Implementa il tuo gioco con flotte gestite di Amazon EC2	198
Ottieni i dati della flotta	202
Aggiungere il FlexMatch matchmaking	203
Gestione dell'hosting con contenitori [Anteprima]	204
Funzionalità principali	204
Utilizzo di flotte di container durante l'anteprima pubblica	205
Come funzionano i container	205
Componenti della flotta di container	205
Architetture comuni	208
Concetti principali	209
Roadmap di sviluppo	212
Integra il tuo gioco con Amazon GameLift	215
Strumenti di integrazione	215
Crea il tuo server di gioco per Linux	217
Prova l'integrazione con una flotta Anywhere	217
Prepara un'immagine del contenitore	219
Crea una cartella di lavoro	219
Crea la tua immagine	221

Invia la tua immagine	229
Progetta una flotta di container	230
Progetta la struttura dei container della tua flotta	231
Imposta i limiti delle risorse	232
Designare contenitori essenziali	234
Configurare le connessioni di rete	235
Imposta i controlli sanitari per i container	239
Imposta le dipendenze del contenitore	240
Configura una flotta di container	240
Crea definizioni di gruppi di contenitori	242
Prima di iniziare	242
Clonare una definizione di gruppo di contenitori	243
Crea una definizione di gruppo di contenitori di replica	243
Crea un file di definizione del contenitore JSON	247
Crea una flotta di container	248
Gestisci le tue flotte di container	254
Visualizzazione delle risorse	254
Aggiornare le risorse	255
Delete resources (Elimina risorse)	255
Scalabilità delle flotte di container	256
Gestione delle risorse di hosting	258
Caricamento di build e script	259
Carica una build	259
Caricare uno script	268
Impostazione dei parchi istanze	273
Guida alla progettazione di parchi istanze	274
Crea una nuova flotta	282
Gestisci le tue flotte	299
Aggiungere un alias a un parco istanze	302
Debug dei problemi del parco istanze	304
Connettiti in remoto alle istanze del parco istanze	308
Scalabilità della capacità di hosting	316
Per gestire la capacità della flotta nella console	317
Imposta limiti di capacità di hosting	317
Impostazione manuale della capacità del parco istanze	319
Ridimensionamento automatico della capacità della flotta	322

Impostazione delle code	329
Progetta una coda	330
Best practice	338
Crea una coda	339
Configurare la notifica degli eventi	343
Tutorial: Code per istanze Spot	346
Gestisci le risorse con AWS CloudFormation	354
Best practice	355
Usare gli AWS CloudFormation stack	356
Aggiornamento delle build	360
Peering VPC	363
Configurazione del peering VPC per un parco istanze esistente	363
Per configurare il peering VPC con un nuovo parco istanze	366
Risoluzione dei problemi di peering VPC	369
Visualizzazione dei dati di gioco	370
Visualizza il tuo GameLift stato su Amazon	370
Visualizza le tue build	372
Dettagli della costruzione	373
Visualizza i tuoi script	373
Dettagli dello script	374
Visualizza le tue flotte	374
Visualizza i dettagli della flotta	375
Informazioni	375
Parametri	376
Eventi	376
Dimensionamento	377
Locations	378
Sessioni di gioco	378
Visualizza le informazioni sul gioco e sul giocatore	378
Informazioni	379
Sessioni giocatore	380
Informazioni sul giocatore	380
Visualizza i tuoi alias	380
Dettagli dell'alias	381
Visualizza le tue code	381
Visualizza i dettagli della coda	382

Monitoraggio di Amazon GameLift	385
Monitoraggio con CloudWatch	386
Dimensioni delle metriche	386
Parametri del parco istanze	387
Metriche della coda	400
Parametri FlexMatch	403
Metriche FleetIQ	407
Registrazione di chiamate API	410
GameLiftInformazioni su Amazon in CloudTrail	410
Comprendere le voci dei file di GameLift registro di Amazon	411
Registrazione dei messaggi del server	413
Registrazione per server personalizzati	414
Registrazione per server in tempo reale	416
Sicurezza	422
Protezione dei dati	423
Crittografia a riposo	424
Crittografia in transito	425
Riservatezza del traffico Internet	425
Gestione dell'identità e degli accessi	425
Destinatari	426
Autenticazione con identità	427
Gestione dell'accesso con policy	430
Come GameLift funziona Amazon con IAM	433
Esempi di policy basate su identità	441
Risoluzione dei problemi	446
Registrazione e monitoraggio con Amazon GameLift	448
Convalida della conformità	449
Resilienza	450
Sicurezza dell'infrastruttura	451
Analisi della configurazione e delle vulnerabilità	452
Best practice di sicurezza	453
Non aprire porte di accesso a Internet	453
Ulteriori informazioni	454
Guide GameLift di riferimento Amazon	455
Riferimento API del servizio (AWSSDK)	455
Configura e gestisci le risorse GameLift di hosting Amazon	455

Inizia sessioni di gioco e unisciti ai giocatori	459
Riferimento ai server in tempo reale	460
Riferimento all'API del client in tempo reale (C#)	461
Riferimento allo script Realtime Servers	475
Riferimento SDK per server	484
Riferimento SDK per server per C++	484
Riferimento SDK per server per C#	561
Riferimento SDK del server per Go	626
Riferimento SDK del server per Unreal Engine	653
Eventi di collocamento delle sessioni di gioco	715
Sintassi degli eventi di posizionamento	715
PlacementFulfilled	716
PlacementCancelled	718
PlacementTimedOut	719
PlacementFailed	720
Prezzo stimato	722
Stima dell'GameLifthosting Amazon	722
GameLiftIstanze Amazon	722
Trasferimento dati in uscita (DTO)	724
Stima Amazon GameLift standalone FlexMatch	725
Quote e regioni supportate	728
Note di rilascio e versioni SDK	729
Versioni SDK	729
Note di rilascio	737
Glossario per AWS	768
.....	dcclxix

Che cos'è Amazon GameLift?

Puoi usare Amazon GameLift per implementare, 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 aiuta a fornire server di gioco ad alte prestazioni e alta affidabilità, scalando dinamicamente l'utilizzo delle risorse per soddisfare la domanda dei giocatori di tutto il mondo.

Usi di Amazon GameLift

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

- Usa i tuoi server di gioco multiplayer personalizzati o usa i server ready-to-go in tempo reale per ospitare i tuoi giochi.
- Esegui risorse di hosting a basso costo utilizzando le istanze Spot di [Amazon Elastic Compute Cloud \(Amazon EC2\)](#).
- Ridimensiona automaticamente la quantità di risorse di hosting di cui il gioco ha bisogno in base all'utilizzo.
- Gestisci le tue risorse di elaborazione Amazon EC2 in un unico posto utilizzando Amazon GameLift FleetIQ.
- Abbina i giocatori nelle partite multigiocatore con Amazon GameLiftFlexMatch.
- Testa in modo iterativo il tuo server di gioco e le tue build client con Amazon. GameLift Anywhere
- Usa il tuo hardware e gestisci tutto in un unico posto con Amazon GameLiftAnywhere.

Tip

Per provare l'hosting di server di GameLift gioco Amazon, consulta [Guida introduttiva ad Amazon GameLift](#).

Inizia a usare le GameLift soluzioni Amazon

GameLiftSoluzioni Amazon per sviluppatori di giochi

- [GameLiftHosting Amazon per server personalizzati](#)
- [GameLiftHosting Amazon con server in tempo reale](#)

- [Amazon GameLift FleetIQ per l'hosting su Amazon EC2](#)
- [Amazon GameLift FlexMatch per il matchmaking](#)
- [Hosting GameLift Anywhere hardware Amazon](#)

GameLiftHosting Amazon per server personalizzati

Amazon GameLift sostituisce il lavoro necessario per ospitare i tuoi server di gioco personalizzati. Le funzionalità di scalabilità automatica ti aiutano a evitare di pagare più risorse del necessario. La scalabilità automatica aiuta anche a garantire che i giochi siano sempre disponibili per i nuovi giocatori con tempi di attesa minimi.

Per ulteriori informazioni sull'GameLift hosting Amazon, consulta [Come GameLift funziona Amazon](#).

Caratteristiche principali

- Usa le funzionalità di GameLift gestione di Amazon, tra cui il ridimensionamento automatico, le code in più sedi e il posizionamento delle sessioni di gioco.
- Distribuire server di gioco per l'esecuzione sui sistemi operativi Amazon Linux o Windows Server.
- Gestire le sessioni di gioco e le sessioni giocatore.
- Configura un monitoraggio dello stato personalizzato per i processi del server per rilevare problemi e risolvere processi con prestazioni scadenti.
- Gestisci le tue risorse di gioco utilizzando AWS CloudFormation modelli per AmazonGameLift.

GameLiftHosting Amazon con server in tempo reale

Usa i server in tempo reale per gestire giochi che non richiedono server di gioco personalizzati. Questa soluzione server leggera fornisce server di gioco che puoi configurare per adattarli al tuo gioco.

Per ulteriori informazioni sull'GameLift hosting Amazon con Realtime Servers, consulta [Integrazione di giochi con Amazon GameLift Realtime Servers](#).

Caratteristiche principali

- Usa le funzionalità di GameLift gestione di Amazon, tra cui il ridimensionamento automatico, le code in più sedi e il posizionamento delle sessioni di gioco.

- Usa le risorse GameLift di hosting di Amazon 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.
- Effettua aggiornamenti in tempo reale alle configurazioni in tempo reale e alla logica del server.

Amazon GameLift FleetIQ per l'hosting su Amazon EC2

Usa Amazon GameLift FleetIQ per lavorare direttamente con le tue risorse di hosting in Amazon EC2 e Amazon EC2 Auto Scaling. Ciò offre il vantaggio delle GameLift ottimizzazioni di Amazon 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 GameLift soluzioni Amazon completamente gestite.

[Per informazioni su come Amazon GameLift FleetIQ funziona con Amazon EC2 ed EC2 Auto Scaling per l'hosting di giochi, consulta la Amazon FleetIQ Developer Guide. GameLift](#)

Caratteristiche principali

- Ottieni un bilanciamento ottimizzato delle istanze Spot utilizzando l'algoritmo FleetIQ.
- Usa le funzionalità di routing dei giocatori per gestire le risorse del tuo server di gioco in modo efficiente e offrire ai giocatori un'esperienza migliore per partecipare ai giochi.
- Ridimensiona automaticamente la capacità di hosting in base all'utilizzo del giocatore.
- Gestisci direttamente le istanze Amazon EC2 direttamente da te. Account AWS
- Usa uno qualsiasi dei formati eseguibili per server di gioco supportati, inclusi Windows, Linux, container e Kubernetes.

Amazon GameLift FlexMatch per il matchmaking

Utilizzalo FlexMatch per creare set di regole personalizzati per definire partite multigiocatore per il tuo gioco. FlexMatch utilizza set di regole per confrontare i giocatori compatibili per ogni partita e offrire ai giocatori l'esperienza multiplayer ideale.

Per ulteriori informazioni FlexMatch, consulta [Cos'è Amazon GameLift FlexMatch?](#)

Caratteristiche principali

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

Hosting GameLift Anywhere hardware Amazon

Usa Amazon GameLift Anywhere per integrare l'hardware in qualsiasi punto del tuo ambiente nel tuo hosting di GameLift giochi Amazon. Puoi integrare Anywhere flotte e flotte EC2 nelle code di matchmaker e sessioni di gioco per gestire il matchmaking e il posizionamento dei giochi sul tuo hardware.

Per ulteriori informazioni sui test con Anywhere, vedere [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#). Per ulteriori informazioni sulla configurazione di una Anywhere flotta, vedere [Configurazione delle GameLift flotte Amazon](#).

Caratteristiche principali

- Esegui test rapidi e iterativi del tuo server di gioco e delle build dei tuoi client.
- Usa i set di GameLift strumenti Amazon per distribuire giochi sul tuo hardware.
- Usa l'hardware più vicino ai tuoi giocatori, ovunque.

Accedere ad Amazon GameLift

Usa questi strumenti per lavorare con AmazonGameLift.

GameLiftSDK Amazon

Gli GameLift SDK Amazon contengono le librerie necessarie per comunicare con Amazon GameLift dai tuoi client di gioco, server di gioco e servizi di gioco. Per ulteriori informazioni, consulta [Supporto allo sviluppo con Amazon GameLift](#).

SDK per client Amazon GameLift Realtime

Realtime Client SDK consente a un client di gioco di connettersi al server Realtime, partecipare a sessioni di gioco e rimanere sincronizzato con gli altri giocatori. Scarica l'[SDK](#) e scopri di più su come effettuare chiamate API con l'[API client Realtime Servers \(C#\)](#).

GameLiftConsole Amazon

Usa [AWS Management Console for Amazon per GameLift](#) gestire le distribuzioni dei giochi, configurare le risorse e tenere traccia delle metriche relative all'utilizzo e alle prestazioni dei giocatori. La GameLift console Amazon fornisce un'alternativa GUI alla gestione delle risorse a livello di codice con `awscli`. AWS Command Line Interface AWS CLI

AWS CLI

Usa questo strumento da riga di comando per effettuare chiamate all'AWS SDK, inclusa l'GameLift API Amazon. Per informazioni sull'utilizzo di AWS CLI, vedere Guida [introduttiva a AWS CLI nella](#) Guida per l'AWS Command Line Interface utente.

Prezzi per Amazon GameLift

Amazon GameLift addebita le istanze in base alla durata di utilizzo e la larghezza di banda in base alla quantità di dati trasferiti. Per un elenco completo degli addebiti e dei prezzi di Amazon GameLift, consulta [GameLift prezzi di Amazon](#).

Per informazioni sul calcolo del costo di hosting dei tuoi giochi o del matchmaking con Amazon GameLift, vedi [Generazione di stime GameLift dei prezzi di Amazon](#), che descrive come utilizzare il [AWS Pricing Calculator](#)

Come GameLift funziona Amazon

Questo argomento tratta i componenti principali dell'hosting di giochi e descrive in che modo Amazon GameLift mette i tuoi server di gioco multiplayer a disposizione dei giocatori.

Sei pronto a preparare il tuo gioco per l'hosting su Amazon GameLift? Dai un'occhiata [Roadmap per l'hosting GameLift gestito di Amazon](#).

Componenti chiave

La configurazione GameLift di Amazon per ospitare il tuo gioco implica l'utilizzo dei seguenti componenti. Il diagramma in [Architettura di gioco con Amazon gestito GameLift](#) visualizza le relazioni tra questi componenti.

- Un server di gioco è il software del server di gioco in esecuzione su una flotta. Carichi la build o lo script del tuo server di gioco su Amazon GameLift e lo comunichi ad Amazon GameLift. Quando usi

Amazon GameLift Anywhere o Amazon GameLift FleetIQ, carichi la build del tuo server di gioco direttamente nella risorsa di elaborazione.

- Una sessione di gioco è una partita in corso con giocatori. Un gioco definisce le caratteristiche di base di una sessione di gioco, come la durata e il numero di giocatori. I giocatori si connettono quindi al server di gioco per partecipare a una sessione di gioco.
- Un client di gioco è il software del gioco in esecuzione su un dispositivo del giocatore. Un client di gioco si connette a un server di gioco tramite servizi di backend per partecipare a una sessione di gioco, in base alle informazioni di connessione che riceve da AmazonGameLift.
- I servizi di backend sono servizi aggiuntivi e personalizzati che gestiscono le attività relative ad AmazonGameLift. Come buona prassi, i tuoi servizi di backend dovrebbero gestire tutte le comunicazioni tra client di gioco e AmazonGameLift.

Hosting di server di gioco

Con AmazonGameLift, puoi ospitare i tuoi server di gioco in tre modi diversi: Amazon gestitoGameLift, Amazon GameLift FleetIQ e Amazon. GameLift Anywhere Per ulteriori informazioni su Amazon GameLift FleetIQ, vedi [Cos'è Amazon GameLift FleetIQ?](#)

È possibile progettare un parco istanze in base alle esigenze di gioco. Per ulteriori informazioni sulla progettazione di una flotta, vedere [Guida alla progettazione del GameLift parco veicoli Amazon](#).

Amazon gestita GameLift

Con Amazon gestitoGameLift, puoi ospitare i tuoi server di gioco su risorse di elaborazione GameLift virtuale Amazon, 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 Anywhere

Con Amazon GameLiftAnywhere, puoi ospitare i tuoi server di gioco su computer che gestisci. Configura le tue risorse di hosting creando una Anywhere flotta che faccia riferimento al tuo computer.

Alias del parco istanze

Un alias è una designazione che puoi trasferire tra flotte, il che lo rende un modo conveniente per avere una posizione generica della flotta. Puoi usare un alias per cambiare client di gioco dall'uso

di una flotta all'altra senza cambiare il tuo client di gioco. Puoi anche creare un alias terminale a cui indirizzare i contenuti.

Sessioni di gioco in corso

Dopo aver distribuito la build del server di gioco su una flotta e dopo che Amazon ha GameLift avviato i processi del server di gioco su ogni istanza, la flotta può ospitare sessioni di gioco. Amazon GameLift avvia nuove sessioni di gioco quando il tuo servizio client di gioco invia una richiesta di collocamento al servizio di backend o ad Amazon. GameLift

Posizionamento delle sessioni di gioco e algoritmo FleetIQ

Le code utilizzano l'algoritmo FleetIQ per selezionare un server di gioco disponibile per ospitare una nuova sessione di gioco. Il componente chiave per il posizionamento delle sessioni di gioco è la coda delle sessioni di GameLift gioco di Amazon. Assegna a una coda di 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 tuo gioco, consulta [Progetta una coda per le sessioni di gioco](#)

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 avviare una nuova sessione di gioco. Il server di gioco risponde alla richiesta e segnala ad Amazon GameLift quando è pronto ad accettare le connessioni dei giocatori. Amazon fornisce GameLift quindi le informazioni di connessione al servizio backend o al servizio client di gioco. I tuoi client di gioco utilizzano queste informazioni per connettersi direttamente alla sessione di gioco e iniziare a giocare.

Dimensionamento della capacità del parco istanze

Quando una flotta è attiva e pronta per ospitare sessioni di gioco, puoi adattare la capacità della flotta per soddisfare la domanda dei giocatori. Ti consigliamo di trovare un equilibrio tra la ricerca rapida di un gioco da parte di tutti i giocatori entranti e la spesa eccessiva in risorse inutilizzate.

Amazon GameLift fornisce uno strumento di scalabilità automatica altamente efficace, oppure puoi impostare manualmente la capacità della flotta. Per ulteriori informazioni, consulta [Scalabilità della capacità di GameLift hosting di Amazon](#).

Auto Scaling

Amazon GameLift offre due metodi di ridimensionamento automatico:

- [Scalabilità automatica basata sugli obiettivi](#)
- [Scalabilità automatica con politiche basate su regole](#)

Altre caratteristiche del dimensionamento

- Protezione delle sessioni di gioco: impedisce ad Amazon GameLift di interrompere le sessioni di gioco che ospitano giocatori attivi durante un evento di ridimensionamento.
- Limiti di scalabilità: controlla l'utilizzo complessivo delle istanze impostando limiti minimi e massimi sul numero di istanze in un parco istanze.
- Sospensione del ridimensionamento automatico: sospendi il ridimensionamento automatico a livello di ubicazione della flotta senza modificare o eliminare le politiche di ridimensionamento automatico.
- Metriche di scalabilità: monitora la cronologia della capacità di una flotta e gli eventi di scalabilità.

Monitoraggio di Amazon GameLift

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

- Dettagli sulla flotta, sulla posizione, sulla sessione di gioco e sulla sessione del giocatore
- Parametri di utilizzo
- Integrità dei processi del server
- Registri delle sessioni di gioco

Per ulteriori informazioni sul monitoraggio in AmazonGameLift, consulta [Monitoraggio di Amazon GameLift](#).

Utilizzo di altre AWS risorse

I tuoi server e le tue applicazioni di gioco possono comunicare con altre AWS risorse. Ad esempio, potresti utilizzare una serie di servizi web per l'autenticazione dei giocatori o i social network. Affinché

i tuoi server di gioco possano accedere alle AWS risorse che Account AWS gestisci, consenti esplicitamente GameLift ad Amazon di accedere alle tue AWS risorse.

Amazon GameLift offre un paio di opzioni per gestire questo tipo di accesso. Per ulteriori informazioni, consulta [Comunica con altre AWS risorse delle tue flotte](#).

Come i giocatori si connettono ai giochi

Una sessione di gioco è un'istanza del tuo gioco in esecuzione su AmazonGameLift. Per giocare al tuo gioco, un giocatore può trovare e partecipare a una sessione di gioco esistente o creare una nuova sessione di gioco e parteciparvi. Un giocatore partecipa creando una sessione di gioco per la sessione di gioco. Se la sessione di gioco è aperta ai giocatori, Amazon GameLift riserva uno spazio per il giocatore e fornisce informazioni sulla connessione. Il giocatore può quindi connettersi alla sessione di gioco e richiedere lo slot prenotato.

Per informazioni dettagliate sulla creazione e la gestione delle sessioni di gioco e delle sessioni dei giocatori con server di gioco personalizzati, consulta [Aggiungi Amazon GameLift al tuo client di gioco](#). Per informazioni sulla connessione dei giocatori ai server in tempo reale, consulta [Integrazione di un client di gioco per server in tempo reale](#).

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

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

Scegli tra più opzioni per configurare il modo in cui Amazon GameLift seleziona le risorse per ospitare nuove sessioni di gioco. Se gestisci flotte in più località, puoi progettare code di sessioni di gioco che inseriscono nuove sessioni di gioco su qualsiasi flotta indipendentemente dalla posizione.

Controlla l'accesso dei giocatori alle sessioni di gioco

Configura le sessioni di gioco per consentire o rifiutare le richieste di partecipazione da parte 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 trasmette i dati delle sessioni di gioco a un server di gioco all'avvio di una nuova sessione di gioco. Amazon GameLift trasmette 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 soluzione 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.

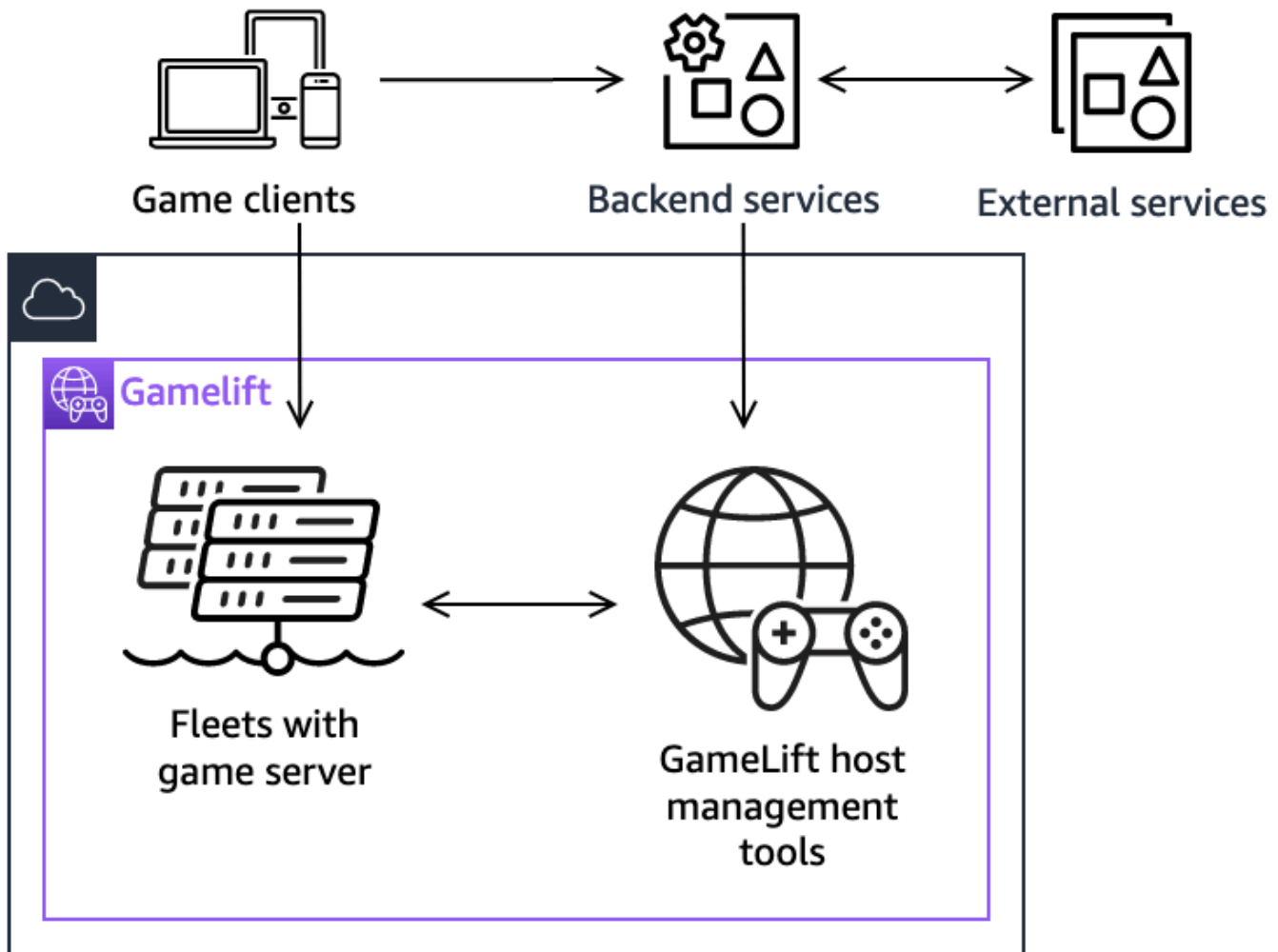
Tieni traccia dei dati di utilizzo del gioco e dei giocatori

Memorizza automaticamente i registri delle sessioni di gioco completate. Puoi configurare l'archiviazione dei log quando integri Amazon GameLift nei tuoi server di gioco. Per ulteriori informazioni, consulta [Registrazione dei messaggi del server in Amazon GameLift](#).

Usa la GameLift console Amazon per visualizzare informazioni dettagliate sulle sessioni di gioco, inclusi i metadati delle sessioni, le impostazioni e i dati delle sessioni del giocatore. Per ulteriori informazioni, consultare [Visualizza i dati sulle sessioni di gioco e dei giocatori](#) e [Parametri](#).

Architettura di gioco con Amazon gestito GameLift

Il diagramma seguente illustra i componenti chiave di un'architettura di gioco ospitata utilizzando la soluzione Amazon GameLift gestita.



I componenti chiave di questa architettura includono quanto segue:

Client di gioco

Per partecipare a un gioco ospitato su Amazon GameLift, il client di gioco deve prima trovare una sessione di gioco disponibile. Il client di gioco cerca sessioni di gioco esistenti, richiede matchmaking o avvia una nuova sessione di gioco comunicando con Amazon GameLift tramite un servizio di backend. Il servizio di backend invia richieste ad Amazon e GameLift, in risposta, riceve informazioni sulla sessione di gioco, che trasmette al client di gioco. Il client di gioco si connette quindi al server di gioco. Per ulteriori informazioni, consulta [Preparazione di giochi per Amazon GameLift](#).

Servizi di backend

Un servizio di backend gestisce la comunicazione tra i client di gioco e Amazon GameLift richiamando le operazioni dell'API del GameLift servizio Amazon nell' AWS SDK. Puoi anche utilizzare i servizi di backend per altre attività specifiche del gioco, come l'autenticazione e l'autorizzazione dei giocatori, l'inventario o il controllo della valuta. Per ulteriori informazioni, consulta [Progetta il tuo servizio client di gioco](#).

Servizi esterni

Il gioco può contare su un servizio esterno, 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.

Server di gioco

Carichi il software del server di gioco su Amazon GameLift e Amazon lo distribuisce GameLift quindi su macchine di hosting per ospitare sessioni di gioco e accettare connessioni tra giocatori. I server di gioco comunicano con Amazon GameLift per avviare sessioni di gioco, convalidare i nuovi giocatori connessi e segnalare lo stato delle sessioni di gioco, delle connessioni dei giocatori e delle risorse disponibili.

I server di gioco personalizzati comunicano con Amazon GameLift utilizzando l'Amazon GameLift Server SDK. I client di gioco si connettono direttamente a un server di gioco dopo aver ricevuto i dettagli di connessione da Amazon GameLift tramite un servizio di backend. Per ulteriori informazioni, consulta [Integra i giochi con server di gioco personalizzati](#).

I server in tempo reale sono server di gioco che eseguono il tuo script personalizzato. Quando si partecipa a una partita, un client di gioco si connette direttamente a un server Realtime utilizzando Realtime Client SDK. Per ulteriori informazioni, consulta [Integrazione di giochi con Amazon GameLift Realtime Servers](#).

Strumenti di gestione dell'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 GameLift strumenti Amazon 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.

Configurazione

Richiedi assistenza per configurare l'utilizzo Account AWS di Amazon GameLift per ospitare i tuoi giochi multiplayer.

Tip

Per provare l'hosting di server di GameLift gioco Amazon, consulta [Guida introduttiva ad Amazon GameLift](#).

Argomenti

- [Configura un Account AWS](#)
- [Supporto allo sviluppo con Amazon GameLift](#)
- [Gestisci i costi di hosting dei giochi](#)
- [Sedi GameLift di hosting Amazon](#)

Configura un Account AWS

Per iniziare a utilizzare Amazon GameLift, 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](#)
- [Crea un utente con accesso amministrativo](#)
- [Gestisci le autorizzazioni degli utenti per Amazon GameLift](#)
- [Configura l'accesso programmatico per gli utenti](#)
- [Configura l'accesso programmatico per il tuo gioco](#)
- [Esempi di autorizzazioni IAM per Amazon GameLift](#)
- [Configura un ruolo di servizio IAM per Amazon GameLift](#)

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 pagina all'indirizzo <https://portal.aws.amazon.com/billing/signup>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso da parte dell'utente root](#).

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo <https://aws.amazon.com/> e selezionando 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 Console](#) come 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, concedi 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 le impostazioni predefinite IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- 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 ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

Per istruzioni, consulta [Creare un set di autorizzazioni](#) nella Guida per l'utente.AWS IAM Identity Center

2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente.AWS IAM Identity Center

Gestisci le autorizzazioni degli utenti per Amazon GameLift

Crea utenti aggiuntivi o estendi le autorizzazioni di accesso agli utenti esistenti, se necessario per le tue GameLift risorse Amazon. 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](#)

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 ai tuoi 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 [Creating a role for a third-party identity provider \(federation\)](#) (Creazione di un ruolo per un provider di identità di terze parti [federazione]) nella Guida per l'utente di IAM.

- Utenti IAM:

- Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina [Creating a role for an IAM user](#) (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di 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 di 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	Utilizza credenziali temporane e per firmare le richieste	Segui le istruzioni per l'interfaccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
(Utenti gestiti nel centro identità IAM)	programmatiche agli AWS CLI AWS SDK o alle API. AWS	<ul style="list-style-type: none">• Per la AWS CLI, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'utente.AWS Command Line Interface• Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento agli AWS SDK e agli strumenti.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Segui le istruzioni in Uso delle credenziali temporanee con AWS risorse nella Guida per l'utente IAM.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	<p>Segui le istruzioni per l'interfaccia che desideri utilizzare.</p> <ul style="list-style-type: none"> • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente. AWS Command Line Interface • Per gli AWS SDK e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli SDK e agli AWS strumenti. • Per le AWS API, consulta 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 accesso. AWS

Configura l'accesso programmatico per il tuo gioco

La maggior parte dei giochi utilizza servizi di backend per comunicare con Amazon GameLift utilizzando gli AWS SDK. Ad esempio, utilizzi un servizio di backend (che agisce per conto dei clienti di gioco) per richiedere sessioni di gioco, inserire giocatori nei giochi e altre attività. Questi servizi richiedono credenziali di accesso e sicurezza programmatiche per autenticare le chiamate alle API dei servizi Amazon GameLift .

Per Amazon GameLift, 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 ad Amazon GameLift. In conformità con le migliori pratiche di sicurezza, i ruoli forniscono un accesso limitato e temporaneo. Puoi utilizzare i ruoli per i 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 le credenziali a lungo termine, che il servizio di backend deve archiviare e utilizzare per comunicare con Amazon. GameLift

Per la sintassi della politica di autorizzazione, consulta. [Esempi di autorizzazioni degli utenti dei 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

Usa la sintassi in questi esempi per impostare le autorizzazioni AWS Identity and Access Management (IAM) per gli utenti che devono accedere alle risorse AmazonGameLift. Per ulteriori informazioni sulla gestione delle autorizzazioni degli utenti, vedere [Gestisci le autorizzazioni degli utenti per Amazon GameLift](#). Quando si gestiscono le autorizzazioni per gli utenti all'esterno dell'IAM Identity Center, è consigliabile assegnare sempre le autorizzazioni ai ruoli o ai gruppi di utenti IAM, non ai singoli utenti.

Se utilizzi Amazon GameLift FleetIQ come soluzione autonoma, consulta [Configurare il tuo Account AWS per Amazon FleetIQ](#). GameLift

Esempi di autorizzazioni di amministratore

Questi esempi forniscono a un utente l'accesso completo alla gestione delle risorse di hosting di GameLift giochi Amazon.

Example Sintassi per le autorizzazioni GameLift delle risorse Amazon

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

```
{
  "Version": "2012-10-17",
  "Statement": {
```

```

    "Effect": "Allow",
    "Action": "gamelift:*",
    "Resource": "*"
  }
}

```

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

L'esempio seguente estende l'accesso a tutte le GameLift risorse e le AWS regioni Amazon che non sono abilitate per impostazione predefinita. Per ulteriori informazioni sulle regioni che non sono abilitate per impostazione predefinita e su come abilitarle, vedere [Gestione Regioni AWS](#) in Riferimenti generali di AWS.

```

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

```

Example Sintassi per GameLift risorse e **PassRole** autorizzazioni Amazon

L'esempio seguente estende l'accesso a tutte le GameLift risorse Amazon e consente a un utente di passare un ruolo di servizio IAM ad AmazonGameLift. Un ruolo di servizio offre ad Amazon una capacità GameLift limitata di accedere ad altre risorse e servizi per tuo conto, come descritto in [Configura un ruolo di servizio IAM per Amazon GameLift](#). Ad esempio, quando risponde a una `CreateBuild` richiesta, Amazon GameLift deve accedere ai tuoi file di build in un bucket Amazon S3. Per ulteriori informazioni sull'`PassRole` azione, consulta [IAM: Passa un ruolo IAM a un AWS servizio specifico](#) nella Guida per l'utente IAM.

```

{
  "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 degli utenti dei giocatori

Questi esempi consentono a un servizio di backend o a un'altra entità di effettuare chiamate API all'GameLiftAPI Amazon. 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 le autorizzazioni di posizionamento delle sessioni di gioco

L'esempio seguente estende l'accesso alle GameLift API di Amazon 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 alle GameLift API di Amazon che gestiscono le attività di FlexMatch matchmaking. FlexMatch associa i giocatori per sessioni di gioco nuove o esistenti e avvia il posizionamento delle sessioni di gioco per i giochi ospitati su AmazonGameLift. Per ulteriori informazioni FlexMatch, consulta [Cos'è Amazon GameLift FlexMatch?](#)

```
{
  "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 alle GameLift API Amazon che creano manualmente sessioni di gioco e sessioni dei 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": "*"
}
```

Configura un ruolo di servizio IAM per Amazon GameLift

Alcune GameLift funzionalità di Amazon richiedono l'estensione dell'accesso limitato alle 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'identità AWS con policy di autorizzazioni che determinano ciò che l'identità può e non può fare in 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 GameLift gestite da Amazon. [Se usi Amazon GameLift FleetIQ per ottimizzare l'hosting di giochi sulle tue istanze Amazon Elastic Compute Cloud \(Amazon EC2\), consulta Configurazione per Amazon FleetIQ. Account AWS GameLift](#)

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

Crea un ruolo IAM personalizzato

Passaggio 1: crea una politica di autorizzazioni.

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

1. Accedi a 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 si crea o si modifica una policy nella AWS Management Console, è possibile 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 CloudFormation modello. Per ulteriori informazioni su AWS CloudFormation, consulta [Informazioni di riferimento sul tipo di risorsa AWS Identity and Access Management](#) nella Guida per l'utente di AWS CloudFormation.

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 di tag in IAM, consulta la sezione [Applicazione di tag alle risorse IAM](#) nella Guida per l'utente di IAM.
11. Seleziona Crea policy per salvare la nuova policy.

Passaggio 2: crea un ruolo che Amazon GameLift possa assumere.

1. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
2. Nella pagina Seleziona un'entità affidabile, 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.

Sintassi della politica di autorizzazione

- Autorizzazioni per Amazon GameLift ad assumere il ruolo di servizio

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

- Autorizzazioni per accedere a 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"
    }
  ]
}
```

Supporto allo sviluppo con Amazon GameLift

Amazon GameLift fornisce un set di SDK che puoi utilizzare con le tue soluzioni di hosting di giochi gestiti. Usa GameLift gli SDK Amazon per aggiungere le funzionalità necessarie ai server di gioco multiplayer, ai client di gioco e ai servizi di gioco che devono interagire con il servizio di GameLift hosting Amazon.

Per le informazioni più recenti sulle versioni di Amazon GameLift SDK e sulla compatibilità degli SDK, consulta [Note di GameLift rilascio di Amazon](#)

Per server di gioco personalizzati

Crea e distribuisce server di gioco personalizzati a 64 bit con l'SDK per GameLift server Amazon. I server di gioco integrati con l'SDK del server e distribuiti per l'hosting possono comunicare con il GameLift servizio Amazon per avviare e gestire le sessioni di gioco. Per informazioni sull'integrazione dell'SDK del server, consulta gli argomenti in [Preparazione di giochi per Amazon GameLift](#)

Sistemi operativi di sviluppo

- Windows

- Linux

Linguaggi di programmazione supportati

Amazon GameLift fornisce l'SDK del server per le seguenti lingue. [Scarica gli SDK del server](#). Per informazioni dettagliate specifiche sulla versione, consultate i file Readme inclusi in ogni pacchetto.

- SDK per server C++
 - [Riferimento SDK](#)
 - [Integrazione SDK](#)
- SDK per server C# (le versioni possono supportare .NET 4 e .NET 6)
 - [Riferimento SDK](#)
 - [Integrazione SDK](#)
- Go
 - [Riferimento SDK](#)
 - [Integrazione SDK](#)

Motori di gioco supportati

Usa SDK specifici per lingua con qualsiasi motore che supporti le librerie C++, C# o Go. Inoltre, Amazon GameLift fornisce questi plugin per motori di gioco: [Scarica i plugin Amazon GameLift](#)

- Unità
 - Il plug-in SDK del server C# per Unity è un plug-in leggero con librerie predefinite che puoi installare utilizzando il gestore di pacchetti Unity. Usa questo plugin con le seguenti versioni di 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.
 - [Integra Amazon GameLift in un progetto Unity](#)
 - Il plug-in standalone per Unity 2021.3 LTS e 2022.3 LTS è un plug-in completo con le librerie SDK C# create per Unity e gli elementi della GUI per la configurazione e la distribuzione di risorse Amazon per l'hosting. GameLift
 - [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#)
 - [Riferimento SDK per GameLift server Amazon per C#](#)

- **Unreal Engine**

Per server di gioco personalizzati

- Il plug-in SDK del server C++ per Unreal è un plug-in leggero composto da codice sorgente C++ Unreal che puoi inserire nelle librerie da utilizzare con le versioni 4, 5 e 5.1 di Unreal Engine.
 - [Integra Amazon GameLift in un progetto Unreal Engine](#)
 - [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)
- Il plug-in standalone per Unreal Engine 5.0, 5.1 e 5.2 è un plug-in completo con le librerie SDK e l'SDK di C++ for Unreal Server. AWS Il plugin è installato nell'editor Unreal, con elementi dell'interfaccia utente e materiali di supporto per la configurazione e la distribuzione delle risorse GameLift Amazon per l'hosting.
 - [Integrazione di giochi con il GameLift plug-in Amazon per Unreal Engine](#)
 - [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)

Sistemi operativi dei server di gioco

Usa l'Amazon GameLift Server SDK per creare server di gioco da eseguire sulle seguenti piattaforme:

- [Windows Server 2016](#)
- [Amazon Linux 2023](#)
- [Amazon Linux 2 \(AL2\)](#)
- [Windows Server 2012](#) (consulta le [GameLift domande frequenti di Amazon per Windows 2012](#))
- [Amazon Linux \(AL1\)](#) (consulta le [GameLift domande frequenti di Amazon per AL1](#))

Per servizi client personalizzati

Crea servizi client personalizzati a 64 bit utilizzando l' AWS SDK con l' GameLift API Amazon. Questo SDK consente ai servizi client di gestire le sessioni di gioco e unire i giocatori ai giochi ospitati su Amazon GameLift. Per iniziare, [scarica l' AWS SDK](#). Per ulteriori informazioni sull'utilizzo dell'SDK con Amazon GameLift, consulta l'[Amazon GameLift API Reference](#).

Per server in tempo reale

Configura e distribuisce server in tempo reale per ospitare i tuoi giochi multiplayer. Per consentire ai client di gioco di connettersi ai server Realtime, usa l'SDK Amazon GameLift Realtime Client. I client di gioco utilizzano questo SDK per scambiare messaggi con un server Realtime e con altri client di

gioco che si connettono al server. Per iniziare, [scarica l'SDK Amazon GameLift Realtime Client](#). Per informazioni sulla configurazione, consulta [Integrazione di un client di gioco per server in tempo reale](#)

Supporto SDK

Il Realtime Client SDK contiene i sorgenti per le seguenti lingue:

- C# (.NET)

Ambienti di sviluppo

Crea 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 server in tempo reale su risorse di hosting che funzionano sulle seguenti piattaforme:

- [Amazon Linux](#)
- [Amazon Linux 2](#)

Gestisci i costi di hosting dei giochi

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/billing/](https://console.aws.amazon.com/billing/). Per ulteriori informazioni sugli strumenti e le risorse per aiutarti a gestire i AWS costi, consulta la [Guida per l'AWS Billing](#). Questa guida può aiutarti a esaminare il consumo di risorse, stabilire l'utilizzo futuro e determinare le tue esigenze di scalabilità.

In particolare, prendi in considerazione questi suggerimenti per aiutarti a gestire i costi dei GameLift servizi Amazon.

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 e altri. Servizi AWS Puoi configurare gli avvisi

per agire in base ai tuoi 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 GameLift flotta Amazon

Utilizza i tag di allocazione AWS dei costi per organizzare e tenere traccia dei costi di hosting dei giochi in base alle flotte Amazon GameLift Amazon EC2 e ad altre risorse EC2. 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 della flotta inutilizzata

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

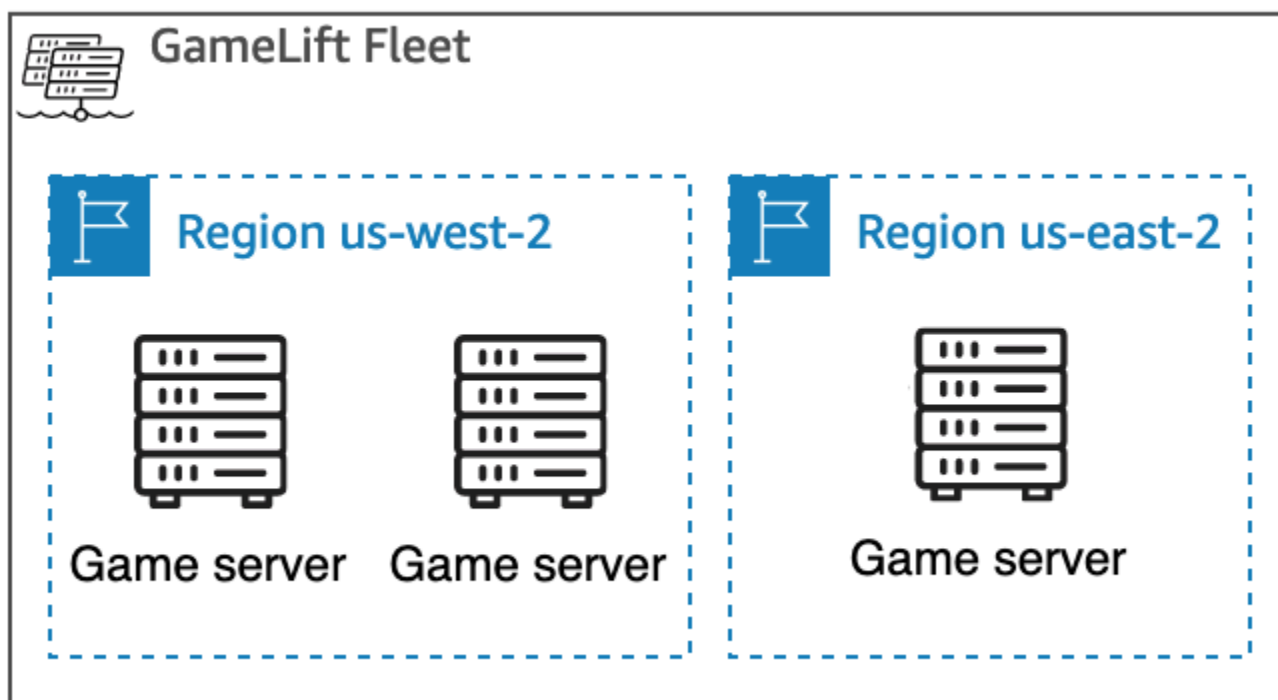
Sedi GameLift di hosting Amazon

Amazon GameLift è disponibile in diverse Regioni AWS Local Zones. Per un elenco completo delle sedi, consulta gli [GameLift endpoint e le quote di Amazon](#) nel. Riferimenti generali di AWS

GameLift Hosting su Amazon

Quando crei una GameLift flotta Amazon, Amazon GameLift crea le risorse della flotta nella tua flotta corrente Regione AWS. Amazon GameLift chiama questa regione la regione di origine della flotta. Per gestire una flotta, accedi alla flotta dalla sua regione di origine.

Le flotte con più sedi distribuiscono istanze in altre località oltre alla regione di origine della flotta. Con le flotte con più sedi, puoi gestire la capacità di ciascuna postazione singolarmente e puoi organizzare sessioni di gioco per località. Le flotte con più sedi possono avere sedi remote in qualsiasi regione o zona locale supportata da Amazon GameLift. Il diagramma seguente mostra una flotta con più sedi con risorse in due regioni. Nel diagramma, la us-west-2 Regione include due server di gioco e la us-east-2 Regione ha un server di gioco.



Se scegli di utilizzare una [flotta con più sedi](#) con istanze in regioni che non sono abilitate per impostazione predefinita, devi abilitare tali regioni nella tua Account AWS. Inoltre, la politica GameLift dell'amministratore di Amazon deve consentire l'`ec2:DescribeRegions`. Per ulteriori informazioni sulle regioni che non sono abilitate per impostazione predefinita e su come abilitarle, consulta [Gestire Regioni AWS](#) in Riferimenti generali di AWS. Per un esempio di politica con le regioni che non sono abilitate per impostazione predefinita, consulta [Esempi di autorizzazioni di amministratore](#).

Important

Per utilizzare le regioni che non sono abilitate per impostazione predefinita, abilitale nel tuoAccount 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.

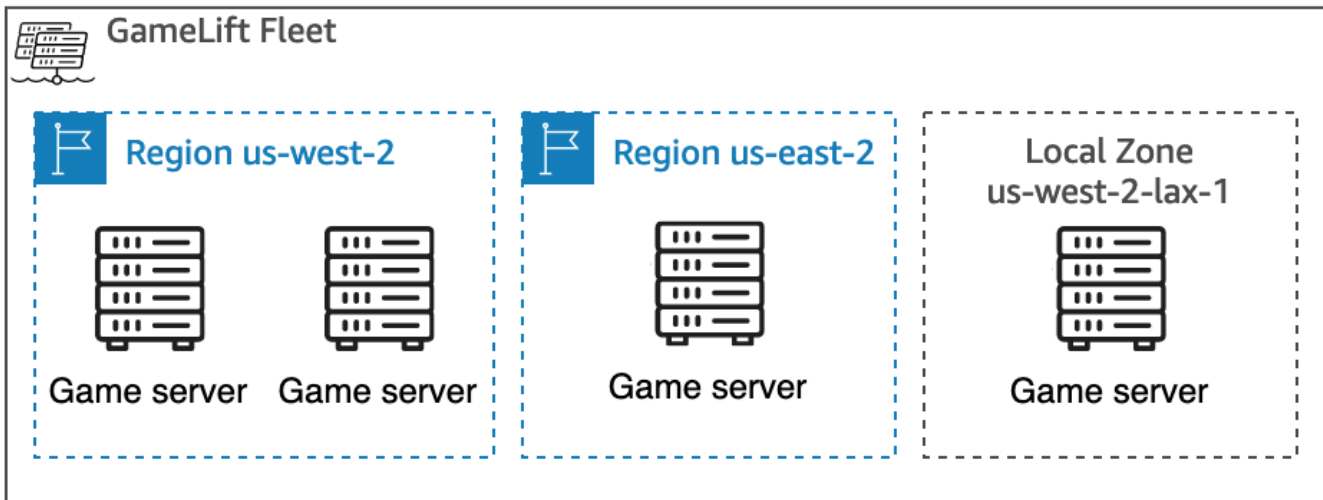
Per il posizionamento delle sessioni di gioco, puoi creare code di sessioni di gioco in qualsiasi posizione GameLift supportata da Amazon. Amazon GameLift colloca le sessioni di gioco dalla posizione in cui hai creato la coda.

Zone locali

Una zona locale è un'estensione di una Regione AWS prossimità geografica ai tuoi utenti. Le Local Zones dispongono di connessioni proprie a Internet. Le Local Zones supportano AWS Direct Connect anche le risorse create in una Local Zone per servire gli utenti locali con comunicazioni a bassa latenza. Per ulteriori informazioni, consulta [Local Zones AWS](#).

Il codice di una zona locale è il codice regionale, seguito da un identificatore che ne indica la posizione fisica. Ad esempio, la us-west-2-lax-1 Local Zone si trova a Los Angeles. Per un elenco delle Local Zones disponibili, vedere [Zone locali disponibili](#).

Amazon GameLift ospita i tuoi giochi in ciascuna delle località che scegli per la tua flotta. Il diagramma seguente mostra una flotta con due server di gioco nella us-west-2 regione, un server di gioco nella us-east-2 regione e un server di gioco nella zona us-west-2-lax-1 locale.



Zone locali disponibili

La tabella seguente elenca le Local Zones disponibili e le relative ubicazioni fisiche.

Local Zone	Ubicazione (area metropolitana)
us-east-1-atl-1	Atlanta
us-east-1-chi-1	Chicago
us-east-1-dfw-1	Dallas
us-east-1-iah-1	Houston
us-east-1-mci-1	Kansas City
us-west-2-den-1	Denver
us-west-2-lax-1	Los Angeles
us-west-2-phx-1	Phoenix

Amazon GameLift Anywhere

Puoi usare Amazon GameLift Anywhere per creare flotte con il tuo hardware e gestire build di gioco, script, server di gioco e client utilizzando Amazon. GameLift Amazon GameLift Anywhere è disponibile in tutte le regioni GameLift supportate da Amazon. Per ulteriori informazioni sulla creazione di una Anywhere flotta e sul test dell'integrazione con i server di gioco, consulta [Crea una GameLift Anywhere flotta Amazon](#) e [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#).

Con Amazon GameLift Anywhere puoi creare posizioni personalizzate che rappresentano la posizione fisica dell'hardware che stai utilizzando per ospitare i tuoi server di gioco GameLift integrati Amazon.

Amazon GameLift FlexMatch

Infatti FlexMatch, puoi ospitare sessioni di gioco generate da partite in qualsiasi luogo supportato da Amazon GameLift. L'effettiva attività di matchmaking si svolge nel luogo in Regione AWS cui hai scelto di creare le tue risorse per il matchmaker. Amazon GameLift indirizza le richieste di abbinamento al matchmaker e le elabora in quella posizione. Per ulteriori informazioni su Amazon GameLift FlexMatch, consulta [What is Amazon GameLift FlexMatch?](#)

[Regioni AWSche supportano FlexMatch le risorse](#)

Amazon GameLift in Cina

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

- [Amazon Web Services in Cina](#)
- [Amazon GameLift](#) (Guida introduttiva ad Amazon Web Services in Cina)

Guida introduttiva ad Amazon GameLift

Ti consigliamo di provare i seguenti esempi prima di utilizzare Amazon GameLift per il tuo gioco. L'esempio di server di gioco personalizzato ti offre esperienza con l'hosting di giochi nella GameLift console Amazon. L'esempio Realtime Servers mostra come preparare un gioco per l'hosting utilizzando Realtime Servers.

Per iniziare a usare Amazon GameLift per il tuo gioco, consulta [Roadmap per l'hosting GameLift gestito di Amazon](#).

Esempio di server di gioco personalizzato

Questo esempio dimostra un gioco personalizzato dal vivo su AmazonGameLift. L'esempio illustra i seguenti passaggi:

- Creazione di una build di gioco di esempio.
- Creazione di una flotta per gestire il server di gioco.
- Connessione al server di gioco dal client di gioco di esempio.
- Revisione delle metriche della flotta e delle sessioni di gioco.

Dopo questi passaggi, puoi avviare più client di gioco e giocare per generare dati di hosting. Quindi, puoi esplorare la GameLift console Amazon per visualizzare le tue risorse di hosting, tenere traccia delle metriche e sperimentare modi per scalare la capacità di hosting.

Per iniziare, accedi alla [GameLiftconsole Amazon](#).

Esempio di gioco Realtime Servers

Questo esempio è un gioco multiplayer completo chiamato Mega Frog Race, con codice sorgente incluso. L'esempio mostra come integrare il tuo client di gioco con Realtime Servers. Puoi anche utilizzare questo gioco di esempio come punto di partenza per sperimentare altre GameLift funzionalità di Amazon come FlexMatch.

Per un tutorial pratico, consulta [Creazione di server per giochi multiplayer per dispositivi mobili con poche righe JavaScript sul blog di AWS for Games](#).

Per il codice sorgente di Mega Frog Race, consulta il [GitHubrepository](#).

Il codice sorgente include le seguenti parti:

- Client di gioco: un codice sorgente per il client di gioco C++, creato in Unity. Il client di gioco riceve 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 API dirette ad AmazonGameLift.
- Script in tempo reale: un file di script sorgente che configura una flotta di server in tempo reale per il gioco. Questo script include la configurazione minima richiesta ai server in tempo reale per comunicare con Amazon GameLift e ospitare giochi.

Roadmap per l'hosting GameLift gestito di Amazon

Questo argomento ti aiuta a scegliere tra le diverse opzioni di GameLift hosting Amazon per il tuo gioco multiplayer basato su sessioni. Gli altri argomenti di questa sezione spiegano come utilizzare Amazon GameLift per il tuo hosting gestito.

Prima di iniziare a preparare il gioco alla produzione, compila il questionario di lancio per iniziare a collaborare con il team di AmazonGameLift.

Argomenti

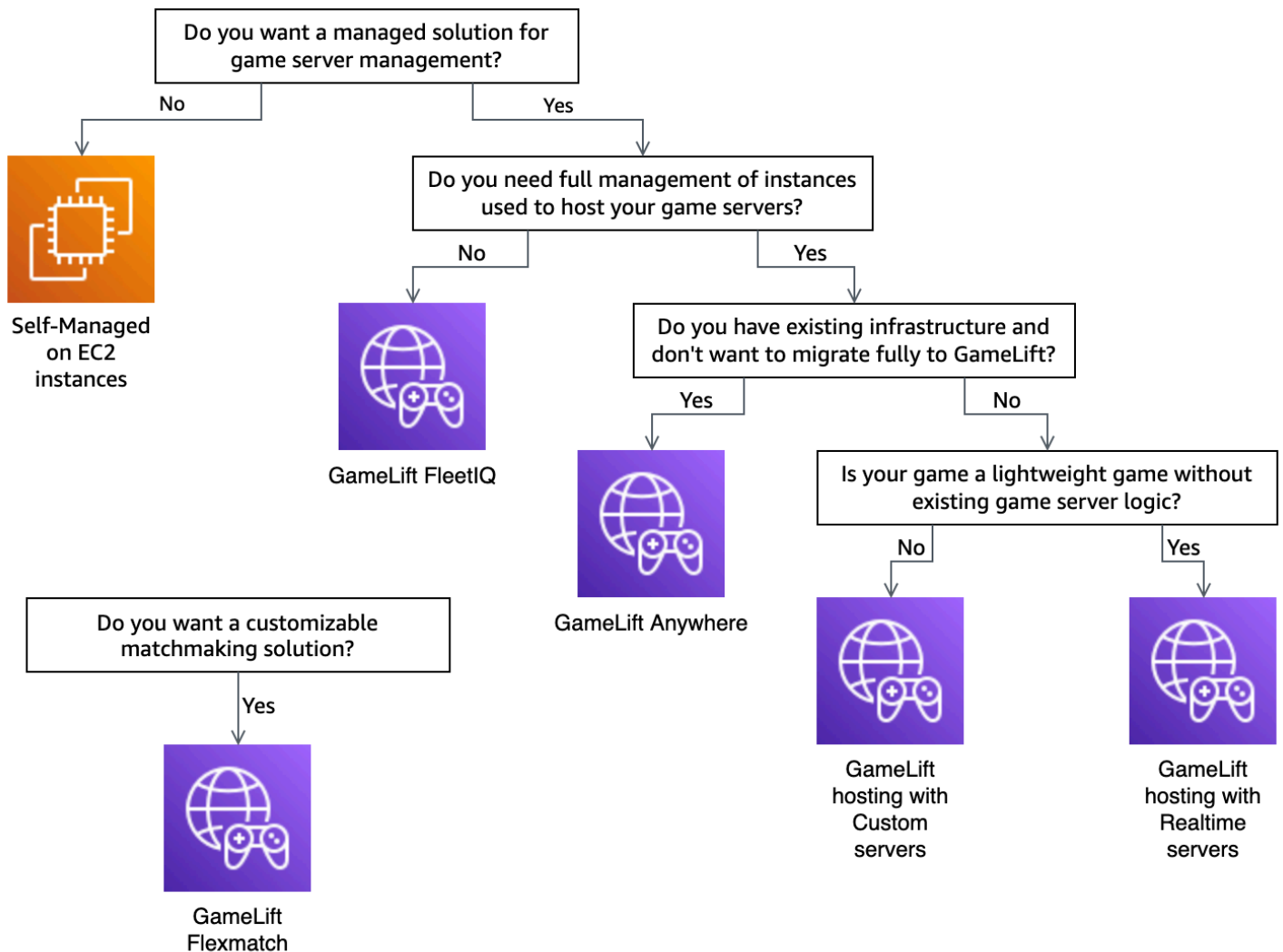
- [Scegli un'opzione di hosting](#)
- [Prepara il tuo gioco per Amazon GameLift](#)
- [Testa la tua integrazione con Amazon GameLift](#)
- [Pianifica e distribuisce le tue risorse Amazon GameLift](#)
- [Progetta il tuo servizio client di gioco](#)
- [Configura le metriche e la registrazione per Amazon GameLift](#)
- [Liste di controllo per il lancio del gioco](#)

Scegli un'opzione di hosting

Il seguente diagramma di flusso pone domande per indirizzarti alla GameLift soluzione Amazon corretta per il tuo caso d'uso.

1. Vuoi una soluzione gestita per la gestione dei server di gioco?
 - Sì, continua con la fase due.
 - No, prendi in considerazione la possibilità di utilizzare server di gioco autogestiti su istanze Amazon EC2.
2. Hai bisogno del pieno controllo delle istanze che ospitano i tuoi server di gioco?
 - Sì, considera Amazon GameLift FleetIQ.
 - No, continua con il passaggio 3.
3. Hai un'infrastruttura esistente che desideri utilizzare con AmazonGameLift?
 - Sì, considera Amazon GameLiftAnywhere.

- No, continua con la fase quattro.
4. Il tuo gioco è leggero senza la logica del server di gioco esistente?
- Sì, prendi in considerazione i server in tempo reale.
 - No: prendi in considerazione i server personalizzati.



Ecco qualche informazione in più su alcune delle opzioni di GameLift hosting Amazon menzionate nel diagramma di flusso:

Amazon GameLift Anywhere

Usa Amazon GameLift Anywhere per ospitare i tuoi giochi sul tuo hardware con i vantaggi degli strumenti di GameLift gestione Amazon. Puoi anche usare le Anywhere flotte per testare i tuoi

server di gioco in modo iterativo. Per ulteriori informazioni, consulta [Crea una GameLift Anywhere flotta Amazon](#).

Amazon gestita GameLift

Esistono due opzioni per l'GameLifthosting gestito di Amazon:

Server personalizzati: Amazon GameLift ospita il tuo server personalizzato che esegue il file binario del tuo server di gioco.

Server in tempo reale: Amazon GameLift ospita il tuo server di gioco leggero.

Amazon GameLift FleetIQ

Nel diagramma di flusso, una migrazione di tipo lift and shift si riferisce a una migrazione in cui non è possibile apportare modifiche all'architettura del gioco. L'utilizzo di Amazon GameLift FleetIQ richiede meno modifiche alla distribuzione esistente e fornisce GameLift strumenti Amazon per la gestione del parco veicoli. Per ulteriori informazioni, consulta la [Amazon GameLift FleetIQ Developer Guide](#).

Se decidi di utilizzare Amazon GameLift Anywhere o Amazon gestitoGameLift, continua a farlo [Prepara il tuo gioco per Amazon GameLift](#).

Prepara il tuo gioco per Amazon GameLift

Questo argomento descrive i passaggi per preparare il gioco multiplayer per l'integrazione con l'GameLifthosting gestito di Amazon. Per preparare il gioco, devi attivare la comunicazione tra il gioco e AmazonGameLift.

Prepara il tuo server di gioco personalizzato

Per avviare e interrompere le sessioni di gioco e per eseguire altre attività, un server di gioco deve essere in grado GameLift di notificare ad Amazon il suo stato. Per attivare la comunicazione con AmazonGameLift, aggiungi del codice al progetto del tuo server di gioco. Per ulteriori informazioni, consulta [Integra i giochi con server di gioco personalizzati](#).

1. Prepara il tuo server di gioco personalizzato per l'hosting su AmazonGameLift.
 - Scarica [Amazon GameLift Server SDK](#) e crealo per il tuo linguaggio di programmazione e motore di gioco preferito.

- Aggiungi codice al progetto del tuo server di gioco per attivare la comunicazione con AmazonGameLift.
2. Prepara il tuo client di gioco per connettersi alle sessioni di gioco GameLift ospitate da Amazon.
 - Aggiungi l'AWSSDK al tuo servizio di backend e al progetto del client di gioco. Per ulteriori informazioni, consulta [Scarica GameLift gli SDK Amazon per i servizi clienti](#).
 - Aggiungi funzionalità per recuperare informazioni sulle sessioni di gioco, inserire nuove sessioni di gioco e riservare spazio ai giocatori in una sessione di gioco.
 - (Facoltativo) Da usare FlexMatch per il matchmaking dei giocatori. Per ulteriori informazioni, consulta [FlexMatch'integrazione con Amazon GameLift hosting](#).

Prepara il tuo server in tempo reale

Amazon GameLift Realtime Servers offre una soluzione server leggera che puoi configurare per adattarla al tuo gioco. Un server in tempo reale offre gli stessi vantaggi che Amazon GameLift offre ai server di gioco, ma con una ridotta personalizzazione dei server di gioco.

Crea uno script in tempo reale per l'hosting su AmazonGameLift.

Gli script in tempo reale contengono la configurazione del server e la logica di gioco personalizzata opzionale. I server in tempo reale sono progettati per avviare e interrompere le sessioni di gioco, accettare le connessioni dei giocatori e gestire le comunicazioni con Amazon GameLift e tra i giocatori di un gioco. Ci sono anche ganci per aggiungere una logica server personalizzata per il tuo gioco. I server in tempo reale utilizzano Node.js e JavaScript. Per ulteriori informazioni, consultare [Creazione di uno script in tempo reale](#) e [Testa la tua integrazione con Amazon GameLift](#).

Testa la tua integrazione con Amazon GameLift

Amazon GameLift supporta l'iterazione rapida durante il test dei server di gioco. Questo argomento illustra i tipi di test disponibili.

Server di gioco personalizzati

Usa Amazon GameLift per integrare l'hardware in qualsiasi punto del tuo ambiente nella tua architettura di hosting di GameLift giochi Amazon. Amazon GameLift Anywhere registra il tuo hardware presso Amazon GameLift in un Anywhere parco macchine, in modo che tu possa testarlo utilizzando il tuo computer di sviluppo locale. Per ulteriori informazioni sui test con Amazon

GameLiftAnywhere, consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#). Per ulteriori informazioni sull'utilizzo di Amazon GameLift Anywhere per ospitare i tuoi giochi con soluzioni locali, consulta [Scelta delle GameLift risorse di elaborazione Amazon](#).

Server in tempo reale

Con Realtime Servers, puoi aggiornare i tuoi script in qualsiasi momento. Quando aggiorni uno script in tempo reale, Amazon GameLift distribuisce la nuova versione alle tue risorse di hosting in pochi minuti. Dopo che Amazon ha GameLift distribuito il nuovo script, tutte le nuove sessioni di gioco utilizzano la nuova versione dello script. Dopo che Amazon GameLift ha distribuito il nuovo script, puoi iniziare immediatamente i test. Per ulteriori informazioni sui server in tempo reale, vedere [Integrazione di giochi con Amazon GameLift Realtime Servers](#)

Pianifica e distribuisci le tue risorse Amazon GameLift

Utilizza i seguenti suggerimenti per pianificare la distribuzione globale GameLift delle risorse Amazon. Per informazioni su dove puoi ospitare i tuoi giochi con AmazonGameLift, consulta [Sedi GameLift di hosting Amazon](#).

Per distribuire le tue GameLift risorse Amazon, completa le seguenti attività:

- Crea un pacchetto e carica il tuo server di gioco su Amazon GameLift o sul tuo hardware. Quando carichi il tuo server su AmazonGameLift, lo carichi solo nella home page Regione AWS della tua flotta. Amazon distribuisce GameLift automaticamente il server in altre sedi del parco veicoli. Per ulteriori informazioni, consulta [Caricamento di build e script su Amazon GameLift](#).
- Progetta e distribuisci una GameLift flotta Amazon per il tuo gioco. Determina il tipo di risorse informatiche da utilizzare, le postazioni in cui eseguire l'installazione, se utilizzare le code e altre opzioni. Per ulteriori informazioni, consulta [Guida alla progettazione del GameLift parco veicoli Amazon](#).
- Crea code per gestire i nuovi posizionamenti delle sessioni di gioco e le strategie delle istanze Spot. Per ulteriori informazioni, consulta [Progetta una coda per le sessioni di gioco](#).
- Usa la scalabilità automatica per gestire la capacità di hosting della tua flotta in base alla domanda prevista dei giocatori. Per ulteriori informazioni, consulta [Scalabilità della capacità di GameLift hosting di Amazon](#).
- Usa le regole del FlexMatch matchmaking per il tuo gioco. Per ulteriori informazioni, consulta [FlexMatch'integrazione con Amazon GameLift hosting](#).

Distribuisci automaticamente le tue risorse Amazon GameLift

Per semplificare la distribuzione globale delle tue GameLift risorse Amazon, ti consigliamo di utilizzare [Infrastructure as code \(IaC\)](#) per definire le risorse. Poiché Amazon GameLift supporta i AWS CloudFormation modelli, puoi impostare i parametri nei modelli per qualsiasi configurazione specifica della distribuzione.

Per gestire l'implementazione dei tuoi AWS CloudFormation stack, ti consigliamo anche di utilizzare strumenti e servizi di integrazione continua e distribuzione continua (CI/CD) come AWS CodePipeline. Questi ti aiutano a eseguire la distribuzione automaticamente o con approvazione ogni volta che crei un file binario per un server di gioco.

Di seguito sono riportati alcuni passaggi comuni della distribuzione GameLift delle risorse Amazon per una nuova versione del server di gioco che puoi automatizzare utilizzando uno strumento o un servizio CI/CD:

- Costruisci e testa il binario del tuo server di gioco.
- Caricare il file binario su Amazon GameLift o sul tuo hardware.
- Implementazione di nuove flotte nella nuova struttura.
- Dopo aver installato le nuove flotte, rimuovi le flotte delle versioni precedenti dalla GameLift coda Amazon e sostituiscile con le nuove flotte.
- Dopo la versione precedente, le flotte terminano con successo tutte le sessioni di gioco, eliminando le AWS CloudFormation pile di quelle flotte.

Puoi anche usare il AWS Cloud Development Kit (AWS CDK) per definire le tue GameLift risorse Amazon. Per ulteriori informazioni su AWS CDK, consulta la [Guida per gli sviluppatori di AWS Cloud Development Kit \(AWS CDK\)](#).

Progetta il tuo servizio client di gioco

Ti consigliamo di implementare un servizio client di gioco che autentichi i tuoi giocatori e comunichi con l'API Amazon. GameLift Implementando un servizio client di gioco personalizzato, puoi:

- Personalizza l'autenticazione per i tuoi giocatori.
- Controlla come Amazon si GameLift abbina e avvia le sessioni di gioco.
- Usa il tuo database dei giocatori per gli attributi dei giocatori come il punteggio di 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 la tua GameLift API Amazon.

Autenticazione dei giocatori

Puoi utilizzare Amazon Cognito e gli ID delle sessioni dei giocatori 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 AWS. Puoi anche utilizzare gli autorizzatori Lambda per una logica di autorizzazione personalizzata con API Gateway.

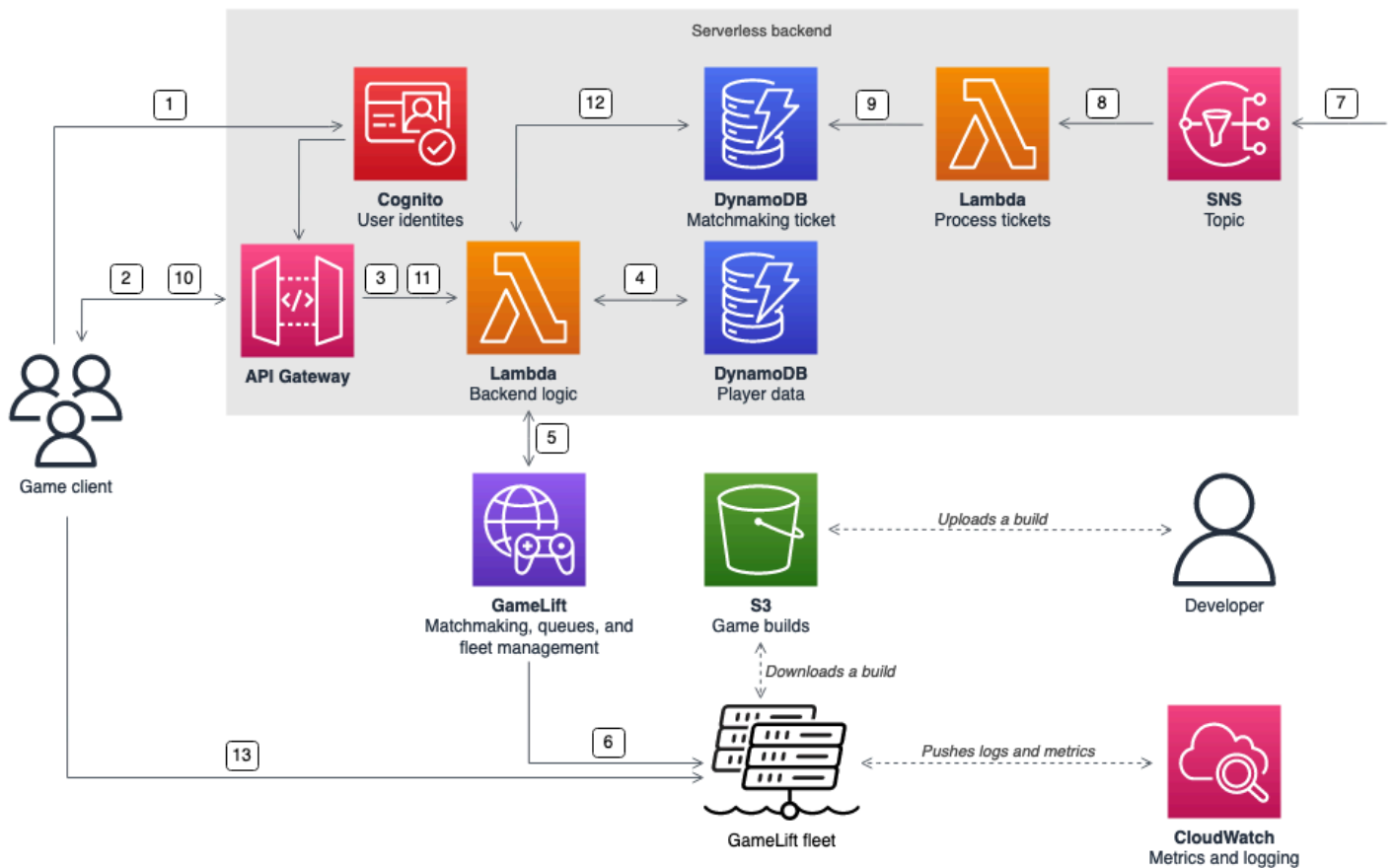
Risorse aggiuntive:

- [Utilizzo dei 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 sui giochi)

Server di sessioni di gioco autonomi con un backend senza server

Utilizzando un'architettura di servizio client senza server, il backend può visualizzare lo stato dei ticket di matchmaking da un database altamente scalabile anziché accedere direttamente all'API Amazon. GameLift

Il diagramma seguente mostra un backend serverless creato per abbinare i giocatori ai giochi in esecuzione sulle flotte Amazon. Servizi AWS GameLift L'elenco seguente fornisce una descrizione per ogni callout numerato nel diagramma. Per provare questo esempio, vedi [Hosting di giochi basato su sessioni multiplayer su 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 contestuali della richiesta.
5. La funzione Lambda richiede una corrispondenza tramite Amazon GameLift FlexMatch matchmaking.
6. FlexMatchabbina un gruppo di giocatori con una latenza adeguata, quindi richiede il posizionamento di una sessione di gioco tramite una GameLift coda Amazon. La coda contiene flotte con una o più Regione AWS sedi.
7. Dopo che Amazon GameLift ha effettuato la sessione in una delle sedi della flotta, Amazon GameLift 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.

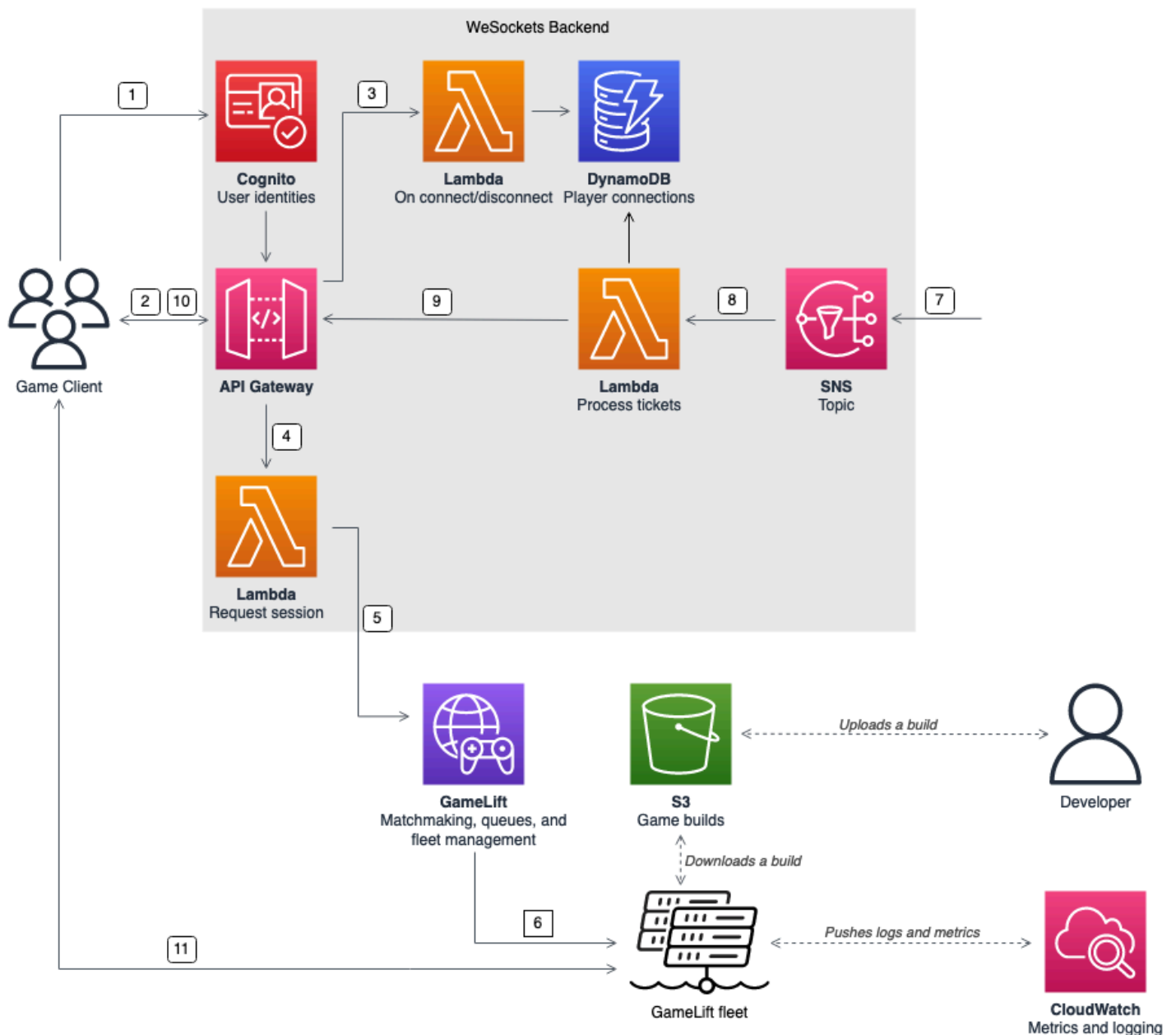
9. Se il ticket di matchmaking è un MatchmakingSucceeded evento, la funzione Lambda scrive il risultato, insieme alla porta e all'indirizzo IP del server di gioco, su una tabella DynamoDB.
- 10 Il client di gioco invia una richiesta firmata ad API Gateway per visualizzare lo stato del ticket di matchmaking in un intervallo specifico.
- 11 API Gateway utilizza una funzione Lambda che controlla 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 al client la porta e l'indirizzo IP del server di gioco, insieme all'ID della sessione del giocatore. Se il ticket non è riuscito, 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 della sessione del giocatore al server di gioco, che quindi convalida l'ID utilizzando l'Amazon GameLift Server SDK.

Server di sessioni di gioco autonomi con un backend WebSocket basato

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 delle API WebSock API Gateway, consulta [Lavorare con le WebSocket API](#).

Il diagramma seguente mostra un'architettura di backend WebSocket basata che utilizza API Gateway e altro Servizi AWS per abbinare i giocatori ai giochi in esecuzione sulle flotte AmazonGameLift. 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 chiama 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 FlexMatch matchmaking.
6. Dopo aver FlexMatch abbinato un gruppo di giocatori, FlexMatch richiede il posizionamento di una sessione di gioco tramite una GameLift coda Amazon.
7. Dopo che Amazon GameLift ha effettuato la sessione in una delle sedi della flotta, Amazon GameLift 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 API Gateway tramite la WebSocket connessione. In questa architettura, il client di gioco non analizza 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 anche l'ID della sessione del giocatore al server di gioco, che quindi convalida l'ID utilizzando l'Amazon GameLift Server SDK.

Configura le metriche e la registrazione per Amazon GameLift

Puoi utilizzare i dati raccolti dai tuoi server di GameLift gioco e dalle tue risorse Amazon per identificare le anomalie. Puoi anche utilizzare le metriche per migliorare le prestazioni.

Le aree chiave da osservare per Amazon GameLift includono:

- Metriche GameLift del servizio Amazon: Amazon GameLift fornisce i CloudWatch parametri Amazon sulle tue risorse, inclusi server di gioco, flotte, code e FlexMatch. Puoi trovare queste metriche nella GameLift console Amazon e nella CloudWatch console. Per ulteriori informazioni sulle GameLift metriche Amazon in CloudWatch, consulta [Monitora Amazon GameLift con Amazon CloudWatch](#).
- Metriche del server di gioco: Amazon non GameLift può accedere alle metriche del tuo server di gioco. Tuttavia, puoi inviare metriche personalizzate CloudWatch direttamente dal tuo server di gioco utilizzando l'CloudWatch agente. Puoi anche utilizzare il ruolo fleet AWS Identity and Access Management (IAM) e l'AWSSDK per inviare le metriche direttamente a CloudWatch. Per un esempio di come configurare le metriche, vedi [Hosting di giochi basato su sessioni multiplayer](#)

su on. AWS GitHub Questo repository include un esempio di configurazione e codice CloudWatch dell'agente per un client C# StatsD.

- Log del server di gioco: per configurare i file di registro del server di gioco sul server di gioco, utilizza la configurazione Amazon GameLift Server SDK. Puoi anche utilizzare Amazon CloudWatch Logs come soluzione di gestione dei log in tempo reale e puoi configurare i log con l'CloudWatch agente. Per ulteriori informazioni, consulta [Registrazione dei messaggi del server in Amazon GameLift](#).

Liste di controllo per il lancio del gioco

Puoi utilizzare queste liste di controllo per convalidare le fasi di implementazione del tuo gioco. Nelle liste di controllo, gli elementi contrassegnati come [Critici] sono fondamentali per il lancio della produzione.

Argomenti

- [Onboarding](#)
- [Test](#)
- [Avvia](#)
- [Dopo il lancio](#)

Onboarding

Usa la seguente lista di controllo per tenere traccia degli elementi necessari per l'onboarding del tuo gioco per l'hosting AmazonGameLift. Gli articoli contrassegnati come [Critici] sono fondamentali per il lancio della produzione.

- [Critico] [Compila il questionario di GameLift onboarding di Amazon nella console Amazon GameLift](#)
- [Critico] [Progetta e implementa un servizio di backend](#) per consentire ai client di gioco di interagire con i tuoi server di gioco.
- [Critico] [Crea ruoli AWS Identity and Access Management \(IAM\)](#) da fornire alle istanze GameLift del server Amazon per l'accesso ad altre AWS risorse.
- [Critico] [Progetta e implementa il failover su altri Regioni AWS](#) moduli FlexMatch e code.
- [Pianifica il lancio delle flotte verso le località di destinazione](#), considerando la coda e la struttura della flotta del gioco.

- [Automatizza la distribuzione](#) utilizzando Infrastructure as code (IaC) con AWS CloudFormation e il AWS Cloud Development Kit (AWS CDK)
- [Raccogli log e analisi](#) utilizzando Amazon CloudWatch e Amazon Simple Storage Service (Amazon S3).

Test

Usa la seguente lista di controllo per tenere traccia degli elementi testati durante lo sviluppo del tuo gioco con l'Amazon GameLift hosting. Gli articoli contrassegnati come [Critici] sono fondamentali per il lancio della produzione.

- [Critico] Completa il questionario di lancio e invia il questionario completato al team di lancio di Amazon. GameLift Puoi trovare il questionario di lancio nella console [Amazon GameLift](#).
- [Critico] [Richiedi aumenti per le quote dei GameLift servizi Amazon](#) e altre Servizio AWS quote in modo che il tuo ambiente reale possa adattarsi alle esigenze di produzione.
- [Critico] Verifica che le porte aperte sulle flotte live corrispondano alla gamma di porte che i tuoi server potrebbero utilizzare.
- [Critico] Chiudere la porta RDP 3389 e la porta SSH 22.
- Sviluppa un piano per la DevOps gestione del tuo gioco. Se utilizzi Amazon CloudWatch Logs o Amazon CloudWatch Custom Metrics, definisci gli allarmi per problemi gravi o critici del parco server. Simula i guasti e testa i runbook.
- [Verifica che il numero di server](#) in esecuzione su un'istanza a pieno utilizzo rientri nelle funzionalità del tipo di istanza del server.
- [Definisci la tua politica di scalabilità](#) in modo che sia inizialmente più prudente e fornisca più capacità inutilizzata di quella che pensi di aver bisogno. Puoi ottimizzare i costi in un secondo momento. Prendi in considerazione l'uso di una politica di scalabilità basata sugli obiettivi con una capacità inattiva del 20%.
- [Usa le regole FlexMatch di latenza](#) per abbinare giocatori geograficamente vicini agli stessi. Regione AWS Verifica come si comporta sotto carico con i dati di latenza sintetici del tuo client di test di carico.
- Prova a caricare l'autenticazione dei tuoi giocatori e l'infrastruttura 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.

- Aumenta il livello del tuo AWS Support piano a Business o Enterprise in modo che AWS possa rispondere alle tue esigenze in caso di problemi o interruzioni.

Avvia

Usa la seguente lista di controllo per tenere traccia degli elementi di lancio del tuo gioco ospitato su AmazonGameLift. Gli articoli contrassegnati come [Critici] sono fondamentali per il lancio della produzione.

- [Critico] [Imposta la politica di protezione della flotta](#) su una protezione completa per tutte le flotte attive, in modo che la riduzione delle dimensioni non interrompa le sessioni di gioco attive.
- [Critico] [Imposta dimensioni massime della flotta](#) sufficientemente elevate da soddisfare al minimo i picchi di domanda previsti. Ti consigliamo di raddoppiare la dimensione massima per una domanda imprevista.
- Incoraggia l'intero team di sviluppo a partecipare all'evento di lancio e monitora il lancio del gioco in una sala di lancio.
- Monitora la latenza e l'esperienza del giocatore.

Dopo il lancio

Usa la seguente lista di controllo per tenere traccia degli articoli post-lancio del tuo gioco ospitati su Amazon. GameLift

- [Ottimizza le regole di scalabilità per ridurre al minimo la capacità inattiva.](#)
- [Modifica FlexMatch le regole](#) o [aggiungi altre posizioni in](#) base ai tuoi requisiti di latenza.
- Ottimizza l'eseguibile del server, poiché la sua efficienza delle prestazioni influisce direttamente sui costi della flotta. Per eseguire più sessioni di gioco con la stessa infrastruttura, aumenta il numero di processi server per istanza.
- [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.

Preparazione di giochi per Amazon GameLift

Per preparare il gioco multigiocatore per l'hosting su Amazon GameLift, configura la comunicazione tra il gioco e Amazon GameLift. Gli argomenti di questa sezione forniscono assistenza dettagliata per integrare il tuo gioco con Amazon GameLift, server di gioco personalizzati e server in tempo reale e per aggiungere matchmaking con FlexMatch

Argomenti

- [Integra i giochi con server di gioco personalizzati](#)
- [Integrazione di giochi con Amazon GameLift Realtime Servers](#)
- [Integrazione di giochi con il GameLift plug-in Amazon per Unity](#)
- [Integrazione di giochi con il GameLift plug-in Amazon per Unreal Engine](#)
- [Ottieni dati sulla flotta per un'istanza Amazon GameLift](#)
- [Aggiungere il FlexMatch matchmaking](#)

Integra i giochi con server di gioco personalizzati

Amazon GameLift fornisce un set completo di strumenti per preparare giochi multiplayer e server di gioco personalizzati da eseguire su AmazonGameLift. Gli GameLift SDK Amazon contengono le librerie necessarie per consentire ai client e ai server di gioco di comunicare con Amazon GameLift. Per ulteriori informazioni sugli SDK e su dove trovarli, consulta [Supporto allo sviluppo con Amazon GameLift](#).

Gli argomenti di questa sezione contengono istruzioni dettagliate su come aggiungere GameLift funzionalità Amazon al client di gioco e al server di gioco prima della distribuzione su AmazonGameLift. Per una tabella di marcia completa per far funzionare il tuo gioco su AmazonGameLift, consulta [Roadmap per l'hosting GameLift gestito di Amazon](#).

Argomenti

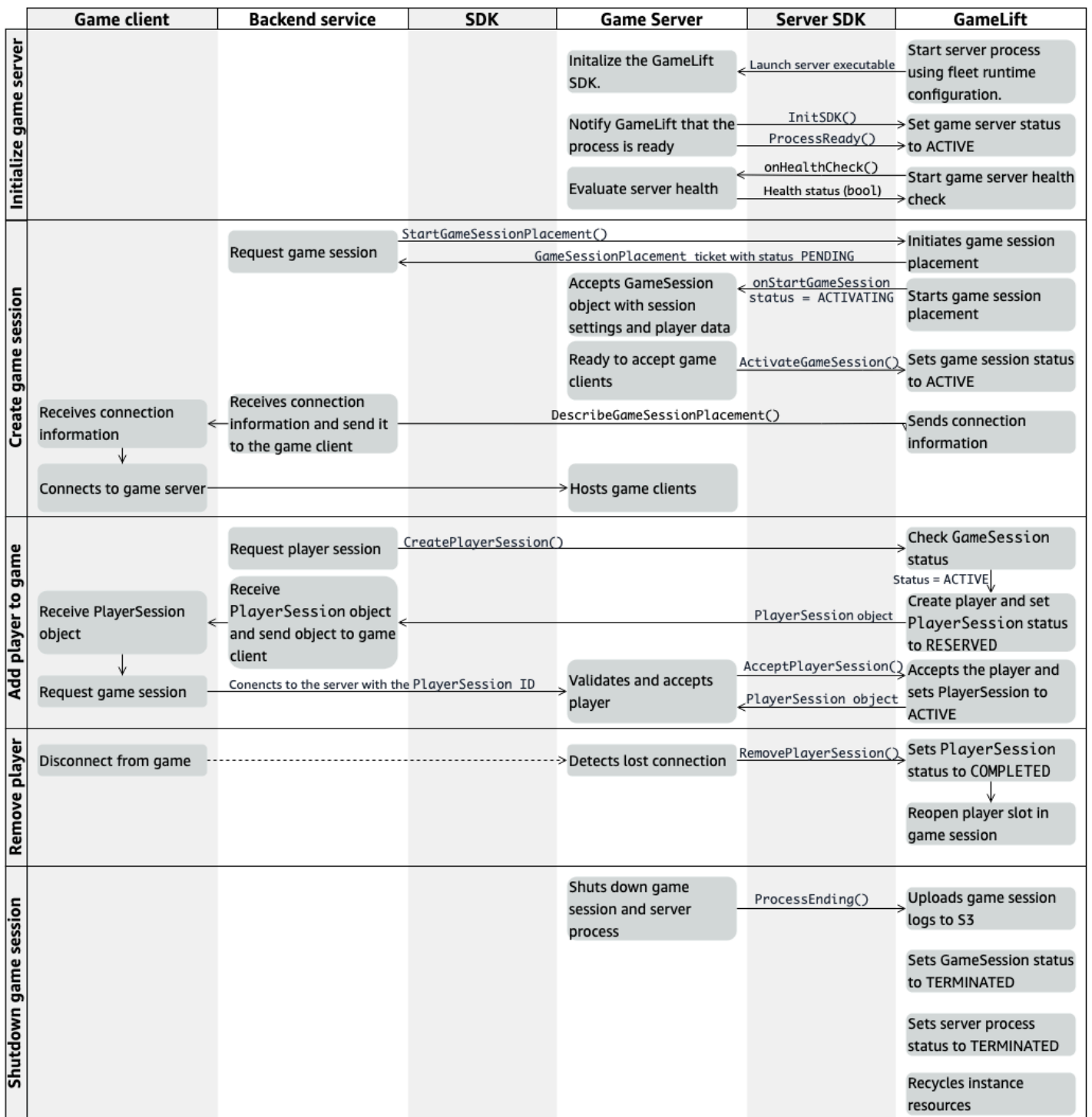
- [Interazioni GameLift tra Amazon e il server client di gioco](#)
- [Integra il tuo server di gioco con Amazon GameLift](#)
- [Integra il tuo client di gioco con Amazon GameLift](#)
- [Motori di gioco e Amazon GameLift](#)
- [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#)

- [Testa la tua integrazione con Amazon GameLift Local](#)

Interazioni GameLift tra Amazon e il server client di gioco

Questo argomento descrive le interazioni tra il client di gioco, un servizio di backend, un server di gioco e AmazonGameLift.

Il diagramma seguente illustra le interazioni tra il client di gioco, il servizio di backend, Amazon GameLift SDK, il server di gioco EC2 gestito, l'SDK del server Amazon e AmazonGameLift. GameLift Per una descrizione dettagliata delle interazioni mostrate, consulta le sezioni seguenti di questa pagina.



Inizializza un server di gioco

I passaggi seguenti descrivono le interazioni che si verificano quando prepari il server di gioco per ospitare sessioni di gioco.

1. Amazon GameLift lancia il server eseguibile su un'istanza Amazon Elastic Compute Cloud (Amazon EC2).
2. Il server di gioco chiama:
 - a. `InitSDK()` per inizializzare l'SDK del server.
 - b. `ProcessReady()` per comunicare la disponibilità della sessione di gioco, le informazioni sulla connessione e la posizione dei file di registro delle sessioni di gioco.

Il processo del server attende quindi una richiamata da Amazon. GameLift

3. Amazon GameLift aggiorna lo stato del processo del server per ACTIVE abilitare il posizionamento delle sessioni di gioco.
4. Amazon GameLift inizia a chiamare il `onHealthCheck` callback e continua a chiamarlo periodicamente mentre il processo del server è attivo. Il processo del server può segnalare lo stato di integrità o meno entro un minuto.

Creare una sessione di gioco

Dopo aver inizializzato il server di gioco, quando crei sessioni di gioco per ospitare i tuoi giocatori si verificano le seguenti interazioni.

1. Il servizio di backend richiama l'operazione SDK. `StartGameSessionPlacement()`
2. Amazon GameLift crea un nuovo `GameSessionPlacement` ticket con status PENDING e lo restituisce al servizio di backend.
3. Il servizio di backend ottiene lo stato di un ticket di collocamento da una coda. Per ulteriori informazioni, consulta [Imposta la notifica degli eventi per il posizionamento della sessione di gioco](#).
4. Amazon GameLift avvia il posizionamento delle sessioni di gioco selezionando una flotta appropriata e cercando un processo server attivo in una flotta con sessioni di 0 gioco. Quando Amazon GameLift individua un processo server, Amazon GameLift effettua le seguenti operazioni:
 - a. Crea un `GameSession` oggetto con le impostazioni della sessione di gioco e i dati del giocatore dalla richiesta di piazzamento con uno ACTIVATING stato.

- b. Richiama il `onStartGameSession` callback sul processo del server. Amazon GameLift trasmette informazioni all'`GameSession` oggetto indicando che il processo del server può configurare la sessione di gioco.
 - c. Cambia il numero di sessioni di gioco del processo server in 1.
5. Il processo del server esegue la funzione di `onStartGameSession` callback. Quando il processo del server è pronto ad accettare le connessioni dei giocatori, chiama `ActivateGameSession()` e attende le connessioni dei giocatori.
6. Amazon GameLift aggiorna l'`GameSession` oggetto con le informazioni di connessione per il processo del server. (Queste informazioni includono l'impostazione della porta che è stata segnalata con `ProcessReady()`.) Amazon modifica GameLift anche lo stato in `ACTIVE`.
7. Il servizio di backend chiama `DescribeGameSessionPlacement()` per rilevare lo stato aggiornato del ticket. Il servizio di backend utilizza quindi le informazioni di connessione per connettere il client di gioco al processo del server e partecipare alla sessione di gioco.

Aggiungere un giocatore a una partita

Questa sequenza descrive il processo di aggiunta di un giocatore a una sessione di gioco esistente. Le sessioni dei giocatori possono anche essere richieste come parte di una richiesta di collocamento di una sessione di gioco.

1. Il servizio di backend richiama l'operazione API del client `CreatePlayerSession()` con un ID di sessione di gioco.
2. Amazon GameLift verifica lo stato della sessione di gioco (deve essere `ACTIVE`) e cerca uno slot per giocatori aperto nella sessione di gioco. Se è disponibile uno slot, Amazon GameLift effettua le seguenti operazioni:
 - a. Crea un nuovo `PlayerSession` oggetto e imposta lo stato su `RESERVED`.
 - b. Risponde alla richiesta del servizio di backend con l'oggetto `PlayerSession`
3. Il servizio di backend collega il client di gioco direttamente al processo del server con l'ID della sessione del giocatore.
4. Il server chiama l'operazione API del server `AcceptPlayerSession()` per convalidare l'ID della sessione del giocatore. Se convalidato, Amazon GameLift passa l'`PlayerSession` oggetto al processo del server. Il processo server accetta o rifiuta la connessione.
5. Amazon GameLift effettua una delle seguenti operazioni:

- a. Se la connessione è accettata, Amazon GameLift imposta lo `PlayerSession` stato su `ACTIVE`.
- b. Se non viene ricevuta alcuna risposta entro 60 secondi dalla `CreatePlayerSession()` chiamata originale del server backend, Amazon GameLift modifica lo `PlayerSession` stato `TIMEDOUT` e riapre lo slot del giocatore nella sessione di gioco.

Rimuovi un giocatore

Quando si rimuovono giocatori da una sessione di gioco per creare spazio per la partecipazione di nuovi giocatori, si verificano le seguenti interazioni.

1. Un giocatore si disconnette dal gioco.
2. Il server rileva la connessione persa e richiama l'operazione `RemovePlayerSession()` API del server.
3. Amazon GameLift modifica lo `PlayerSession` stato `COMPLETED` e riapre lo slot del giocatore nella sessione di gioco.

Chiudi la sessione di gioco

Questa sequenza di interazioni si verifica quando un processo del server interrompe la sessione di gioco corrente.

1. Il server chiude la sessione di gioco e il server.
2. Il server chiama `ProcessEnding()` AmazonGameLift.
3. Amazon GameLift fa quanto segue:
 - a. Carica i registri delle sessioni di gioco su Amazon Simple Storage Service (Amazon S3).
 - b. Cambia lo `GameSession` stato in `TERMINATED`.
 - c. Cambia lo stato del processo del server in `TERMINATED`.
 - d. Ricicla le risorse delle istanze.

Integra il tuo server di gioco con Amazon GameLift

Dopo che il tuo server di gioco personalizzato è stato distribuito e funzionante su GameLift istanze Amazon, deve essere in grado di interagire con Amazon GameLift (e potenzialmente

con altre risorse). Questa sezione descrive come integrare il software del tuo server di gioco con AmazonGameLift.

Note

Queste istruzioni presuppongono che tu abbia creato un progetto di server di gioco Account AWS e che tu disponga di un progetto di server di gioco esistente.

Gli argomenti di questa sezione descrivono come gestire le seguenti attività di integrazione:

- Stabilisci una comunicazione tra Amazon GameLift 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 affinché il software del tuo server di gioco interagisca 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 al tuo server di gioco](#)
- [Comunica con altre AWS risorse delle tue flotte](#)

Aggiungi Amazon GameLift al tuo server di gioco

Il tuo server di gioco personalizzato deve comunicare con AmazonGameLift, perché ogni processo del server di gioco deve essere in grado di rispondere agli eventi GameLift avviati da Amazon. Il tuo server di gioco deve inoltre tenere Amazon GameLift informata sullo stato del processo del server e sulle connessioni dei giocatori. Per ulteriori informazioni su come il server di gioco, il servizio di backend, il client di gioco e Amazon GameLift collaborano per gestire l'hosting dei giochi, consulta [Interazioni GameLift tra Amazon e il server client di gioco](#).

Per preparare il tuo server di gioco a interagire con AmazonGameLift, aggiungi Amazon GameLift Server SDK al progetto del server di gioco e integra le funzionalità descritte in questo argomento. Il Server SDK è disponibile in diverse lingue. Per ulteriori informazioni su Amazon GameLift Server SDK, consulta [Supporto allo sviluppo con Amazon GameLift](#).

Riferimenti API SDK del server:

- [Riferimento Amazon GameLift Server SDK 5.x per C++](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)
- [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)

Inizializza il processo del server

Aggiungi codice per stabilire una comunicazione con Amazon GameLift e segnalare che il processo del server è pronto per ospitare una sessione di gioco. Questo codice deve essere eseguito prima di qualsiasi GameLift codice Amazon.

1. Inizializza il client Amazon GameLift API `InitSdk()` chiamando. Per inizializzare un processo server su una risorsa di GameLift Anywhere elaborazione Amazon, chiama `InitSdk()` con quanto segue: `ServerParameters`
 - L'URL del websocket utilizzato per connettersi al server di gioco.
 - L'ID del processo utilizzato per ospitare il tuo server di gioco.
 - L'ID del computer che ospita i processi del server di gioco.
 - L'ID del GameLift parco veicoli contenente i tuoi dati di GameLift Anywhere calcolo Amazon.
 - Il token di autorizzazione generato dall'operazione Amazon.

[GetComputeAuthToken](#)

Note

[Per inizializzare un server di gioco su un'istanza Amazon EC2 GameLift gestita da Amazon, costruisci il tuo `ServerParameters` utilizzando il costruttore predefinito `InitSDK\(\)` \(C++\) \(C++\) \(Unreal\)](#) Amazon GameLift configura l'ambiente di elaborazione e si connette automaticamente ad Amazon GameLift per te.

2. Notifica ad Amazon GameLift che un processo del server è pronto per ospitare una sessione di gioco. Chiama `ProcessReady()` ([C++](#)) ([C#](#)) ([Unreal](#)) (C++) con le seguenti informazioni. (Nota che dovresti chiamare `ProcessReady()` solo una volta per processo del server).

- Il numero di porta utilizzato dal processo del server. Il servizio di backend fornisce il numero di porta e un indirizzo IP ai client di gioco per connettersi al processo del server e partecipare a una sessione di gioco.
- La posizione dei file, come i registri delle sessioni di gioco, che desideri che Amazon GameLift conservi. Il processo del server genera questi file durante una sessione di gioco. Vengono temporaneamente archiviati nell'istanza in cui è in esecuzione il processo del server e vengono persi quando l'istanza si spegne. Tutti i file che elenchi vengono caricati su AmazonGameLift. Puoi accedere a questi file tramite la [GameLiftconsole Amazon](#) o chiamando l'operazione Amazon GameLift API [GetGameSessionLogUrl\(\)](#).
- I nomi delle funzioni di callback che Amazon GameLift può richiamare sul tuo processo server. Il tuo server di gioco deve implementare queste funzioni. [Per ulteriori informazioni, vedere \(C++\) \(C#\) \(Unreal\) \(C++\)Unreal](#).
 - (Facoltativo)`onHealthCheck`: Amazon GameLift chiama questa funzione regolarmente per richiedere un rapporto sullo stato di salute dal server.
 - `onStartGameSession`— Amazon GameLift chiama questa funzione in risposta alla richiesta del client [CreateGameSession\(\)](#).
 - `onProcessTerminate`— Amazon GameLift forza l'interruzione del processo del server, lasciandolo chiudere correttamente.
 - (Facoltativo)`onUpdateGameSession`: Amazon GameLift fornisce un oggetto della sessione di gioco aggiornato al server di gioco o fornisce un aggiornamento sullo stato di una richiesta di backfill della partita. La funzione di [FlexMatchbackfill](#) richiede questo callback.

Puoi anche configurare un server di gioco per accedere in modo sicuro alle AWS risorse che possiedi o che controlli. Per ulteriori informazioni, consulta [Comunica con altre AWS risorse delle tue flotte](#).

(Facoltativo) Stato dei processi del server di report

Aggiungi codice al tuo server di gioco per implementare la funzione di callback. `onHealthCheck()` Amazon GameLift richiama periodicamente questo metodo di callback per raccogliere i parametri sanitari. Per implementare questa funzione di callback, procedi come segue:

- Valuta lo stato di salute del processo del server. Ad esempio, potresti segnalare il processo del server come non integro se le dipendenze esterne non sono riuscite.

- Completare la valutazione dello stato e rispondere alla richiamata entro 60 secondi. Se Amazon GameLift non riceve una risposta in quel lasso di tempo, considera automaticamente il processo del server non sano.
- Restituisce un valore booleano: vero per sano, falso per malsano.

Se non implementi un callback per il controllo dello stato, Amazon GameLift considera il processo del server integro a meno che il server non risponda.

Amazon GameLift utilizza l'integrità dei processi del server per porre fine a processi non sani e ripulire le risorse. Se un processo del server continua a essere segnalato come non integro o non risponde per tre controlli di integrità consecutivi, Amazon GameLift potrebbe interrompere il processo e avviarne uno nuovo. Amazon GameLift raccoglie le metriche sullo stato dei processi dei server di un parco veicoli.

(Facoltativo) Ottieni un certificato TLS

Se il processo server è in esecuzione su un parco macchine 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 tra server client. Una copia del certificato viene archiviata nell'istanza. [Per ottenere la posizione del file, chiama `GetComputeCertificate\(\)`\(C++\) \(C#\) \(Unreal\) \(C++\)Unreal\).](#)

Avvia una sessione di gioco

Aggiungere il codice per implementare la funzione di richiamata `onStartGameSession`. Amazon GameLift richiama questo callback per avviare una sessione di gioco sul server.

La `onStartGameSession` funzione accetta un [GameSession](#) oggetto 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 Amazon GameLift quando una nuova sessione di gioco è pronta ad accettare giocatori. [Chiama l'operazione dell'API del server `ActivateGameSession\(\)`\(C++\) \(C#\) \(Unreal\) \(C+](#)

[+\)Unreal](#)). In risposta a una chiamata andata a buon fine, Amazon GameLift modifica lo stato della sessione di gioco in `INACTIVE`.

(Facoltativo) Convalida un nuovo giocatore

Se stai monitorando lo stato delle sessioni dei giocatori, aggiungi un codice per convalidare un nuovo giocatore quando si connette a un server di gioco. Amazon GameLift tiene traccia dei giocatori attuali e degli slot disponibili per le sessioni di gioco.

Per la convalida, un client di gioco che richiede l'accesso alla sessione di gioco deve includere un ID della sessione del giocatore. Amazon genera GameLift automaticamente questo ID quando un giocatore chiede di partecipare a una partita utilizzando [StartGameSessionPlacement\(\)](#) o [StartMatchmaking\(\)](#). La sessione del giocatore riserva quindi uno spazio libero in una sessione di gioco.

Quando il processo del server di gioco riceve una richiesta di connessione al client di gioco, chiama `AcceptPlayerSession()` ([C++](#)) ([C#](#)) ([Unreal](#)) (`C++`) della sessione del giocatore. In risposta, Amazon GameLift verifica che l'ID della sessione del giocatore corrisponda a uno slot aperto riservato nella sessione di gioco. Dopo che Amazon ha GameLift 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 non convalida l'ID della sessione del giocatore, il processo del server nega la connessione.

(Facoltativo) Segnala la fine di una sessione con un giocatore

Se stai monitorando lo stato delle sessioni dei giocatori, aggiungi un codice per avvisare Amazon GameLift quando un giocatore abbandona la sessione di gioco. Questo codice deve essere eseguito ogni volta che il processo del server rileva un'interruzione della connessione. Amazon GameLift utilizza 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 API del server `RemovePlayerSession()` ([C++](#)) ([C#](#)) ([Unreal](#)) (`C++`) della sessione del giocatore corrispondente.

Termina una sessione di gioco

Aggiungi codice alla sequenza di spegnimento del processo del server per avvisare Amazon GameLift quando termina una sessione di gioco. Per riciclare e aggiornare le risorse di hosting, Amazon GameLift interrompe i processi del server al termine della sessione di gioco.

[All'avvio del codice di spegnimento del processo server, chiama l'operazione API del server](#)

[ProcessEnding\(\) \(C++\) \(C#\) \(Unreal\) \(C++\) \(C#\) \(Unreal\)](#). Questa chiamata avvisa Amazon GameLift che il processo del server si sta spegnendo. Amazon GameLift modifica 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.

Rispondere a una notifica di arresto del processo del server

Aggiungi codice per interrompere il processo del server in risposta a una notifica di Amazon GameLift. Amazon GameLift invia questa notifica quando il processo del server segnala costantemente problemi di integrità o se l'istanza in cui è in esecuzione il processo del server viene interrotta. Amazon GameLift può interrompere un'istanza come parte di un evento di riduzione della capacità o in risposta all'interruzione di un'istanza Spot.

Per gestire una notifica di spegnimento, apporta le seguenti modifiche al codice del server di gioco:

- Implementare la funzione di richiamo `onProcessTerminate()`. Questa funzione dovrebbe richiamare il codice che arresta il processo del server. Quando Amazon GameLift richiama questa operazione, le interruzioni delle istanze Spot forniscono un preavviso di due minuti. Questo avviso dà al server il tempo necessario per disconnettere i giocatori con garbo, preservare i dati sullo stato del gioco ed eseguire altre attività di pulizia.
- Richiama l'operazione dell'API del server `GetTerminationTime()` ([C++](#)) ([C#](#)) ([Unreal](#)) ([C++](#)) ([C#](#)) ([Unreal](#)) dal codice di spegnimento del server di gioco. Se Amazon GameLift ha effettuato una chiamata per interrompere il processo del server, `GetTerminationTime()` restituisce il tempo di terminazione stimato.
- [All'inizio del codice di spegnimento del server di gioco, chiama l'operazione API del server ProcessEnding\(\) \(C++\) \(C#\) \(Unreal\) \(C++\) \(C#\) \(Unreal\)](#). Questa chiamata notifica ad Amazon GameLift che il processo del server si sta spegnendo e Amazon GameLift 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 un server di gioco da distribuire su GameLift flotte Amazon, 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 gestisce le tue flotte di hosting di giochi, devi concedere ad Amazon un accesso GameLift limitato a queste risorse e servizi.

Alcuni scenari di esempio includono:

- Utilizza un CloudWatch agente Amazon per raccogliere metriche, log e tracce da flotte e flotte EC2 gestite Anywhere
- Invia i dati di log delle istanze ad Amazon Logs. CloudWatch
- 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 supporta 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 questa fase, crei un ruolo IAM, con una serie di autorizzazioni per controllare l'accesso alle tue AWS risorse e una politica di fiducia che concede ad Amazon GameLift i diritti di utilizzo delle autorizzazioni del ruolo.

Per istruzioni su come configurare il ruolo IAM, consulta [Configura un ruolo di servizio IAM per Amazon GameLift](#). Quando crei la politica delle autorizzazioni, scegli i servizi, le risorse e le azioni specifici con cui le tue applicazioni devono lavorare. Come procedura ottimale, limita il più possibile l'ambito delle autorizzazioni.

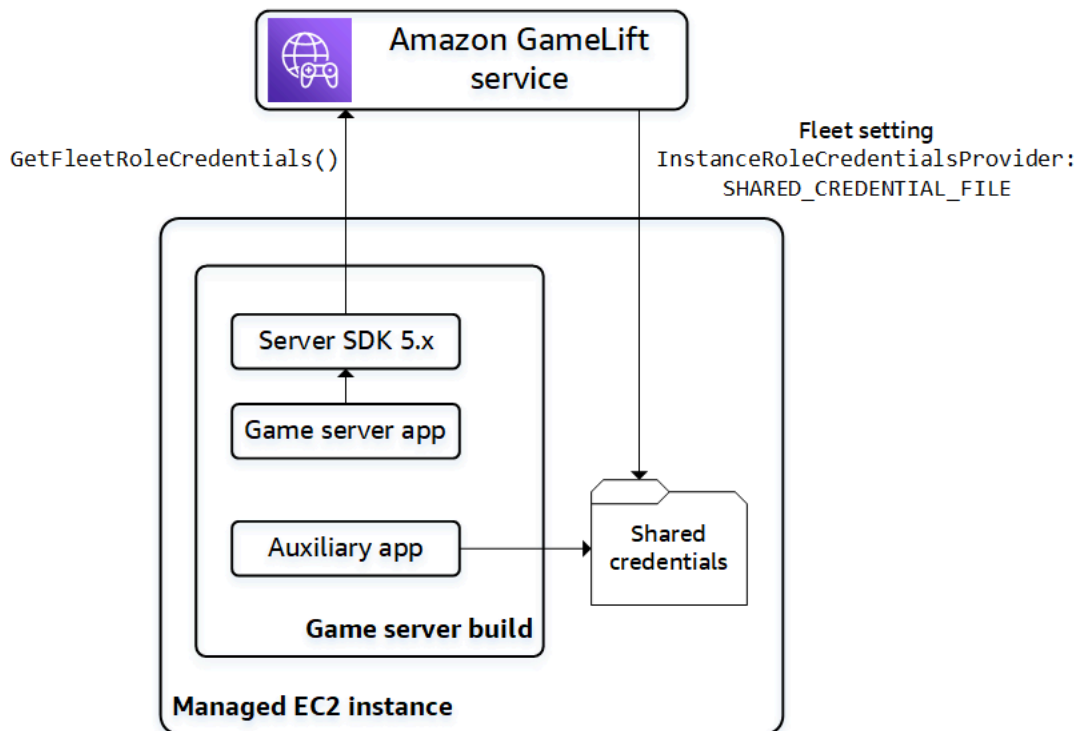
Dopo aver creato il ruolo, prendi nota dell'Amazon Resource Name (ARN) del ruolo. È necessario il ruolo ARN durante la creazione della flotta.

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 utilizzata dal gioco per comunicare con Amazon. GameLift

	Applicazioni per server di gioco	Altre applicazioni
Utilizzo della versione 5.x dell'SDK del server	Richiama il metodo SDK del server <code>GetFleetRoleCredentials()</code> dal codice del server di gioco.	Aggiungi codice all'applicazione per estrarre le credenziali da un file condiviso sull'istanza fleet.
Utilizzando la 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 Amazon GameLift Server SDK 5.x, chiama `GetFleetRoleCredentials` ([C++](#)) ([C#](#)) ([Unreal](#)) 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 genera 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 aggiorna GameLift continuamente le credenziali temporanee.

È necessario configurare una flotta per generare il file di credenziali condivise al momento della 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 i parametri per le istanze della flotta Amazon GameLift

Se desideri utilizzare un CloudWatch agente Amazon per raccogliere metriche, log e tracce dalle tue GameLift flotte Amazon, 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 GameLift della flotta Amazon 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 [Utilizzo di credenziali temporanee con le risorse AWS](#) nella Guida per l'utente IAM.

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](#)— Per richiedere ad Amazon di GameLift generare 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' GameLift istanza Amazon 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 GameLift flotta Amazon 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 semplifica il processo di configurazione delle connessioni peering VPC per i tuoi 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](#)

Integra il tuo client di gioco con Amazon GameLift

Gli argomenti di questa sezione descrivono le GameLift funzionalità gestite di Amazon che puoi aggiungere a un servizio di back-end. Un servizio di backend gestisce le seguenti attività:

- Richiede informazioni sulle sessioni di gioco attive da AmazonGameLift.
- Inserisce un giocatore in una sessione di gioco esistente.
- Crea una nuova sessione di gioco e unisce i giocatori.
- Modifica i metadati per una sessione di gioco esistente.

Per ulteriori informazioni su come i client di gioco interagiscono con Amazon GameLift e i server di gioco in esecuzione su AmazonGameLift, consulta [Interazioni GameLift tra Amazon e il server client di gioco](#).

Prerequisiti

- Un Account AWS.
- Una build del server di gioco caricata su AmazonGameLift.
- Una flotta per ospitare i tuoi giochi.

Argomenti

- [Aggiungi Amazon GameLift al tuo client di gioco](#)
- [Genera gli ID dei giocatori](#)

Aggiungi Amazon GameLift al tuo client di gioco

Integra Amazon GameLift nei 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 si trova 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 GameLift ospitato su Amazon, consulta la [Amazon GameLift FlexMatch Developer Guide](#).

Configura Amazon GameLift su un servizio di backend

Aggiungi codice per inizializzare un GameLift client Amazon e memorizzare le impostazioni chiave. Questo codice deve essere eseguito prima di qualsiasi codice dipendente da Amazon GameLift.

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 quando si contatta Amazon. GameLift La regione identifica l'insieme di risorse distribuite (flotte,

code e matchmaking) 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. Inizializza un GameLift client Amazon. 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 gli ID dei giocatori](#).
4. Raccogli e archivia le seguenti informazioni:
 - Flotta Target: molte richieste di GameLift API Amazon 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 ad Amazon GameLift devono fornire le credenziali per Account AWS l'host del gioco. Acquisisci 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 degli utenti dei giocatori, aggiungi il codice per assumere il ruolo prima di chiamare un' API GameLift Amazon. 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 AWS SDK e agli 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 gli AWS SDK nella Guida per l'utente 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 [SearchGameSessions](#) di 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 [GameSession](#) oggetto 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: `ValutaCreationTime`.
- 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 [StartGameSessionPlacement](#) per inserire una richiesta per una nuova sessione di gioco in una coda. Per utilizzare questa operazione, crea una coda. Ciò determina dove Amazon GameLift colloca la nuova sessione di gioco. Per ulteriori informazioni sulle code e su come utilizzarle, consulta [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#).

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 utilizza queste informazioni per collocare la nuova sessione di gioco su una flotta che offre l'esperienza di gioco ideale per i giocatori.

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 [CreateGameSession](#) per 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 che Amazon ha GameLift creato la nuova sessione di gioco e [GameSession](#) restituito un oggetto, puoi unirvi altri giocatori.

Quando utilizzi 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 funzione di protezione GameLift delle risorse di Amazon 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, consulta [Come i giocatori si connettono ai giochi](#).

Esistono due modi per creare nuove sessioni con i giocatori:

- [StartGameSessionPlacement](#) Utilizzalo per riservare slot per uno o più giocatori nella nuova sessione di gioco.
- Riserva gli slot per uno o più giocatori utilizzando [CreatePlayerSession](#) [CreatePlayerSessions](#) con un ID di sessione di gioco.

Amazon verifica GameLift innanzitutto che la sessione di gioco accetti nuovi giocatori e che disponga di slot per giocatori. In caso di successo, Amazon GameLift riserva uno slot per il giocatore, crea la nuova sessione giocatore e restituisce un [PlayerSession](#) oggetto. 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 gli ID dei giocatori](#).

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 passa questo oggetto GameLift 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 del server comunicano direttamente senza coinvolgere Amazon GameLift. Il server mantiene la comunicazione con Amazon GameLift per 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 `Easy` poi inizia la partita. Il tuo client di gioco richiede una nuova sessione di gioco GameLift da Amazon utilizzando una delle due `CreateGameSession` opzioni `StartGameSessionPlacement` o come spiegato nelle sezioni precedenti. Nella richiesta, il client passa questo: `{"Key": "Difficulty", "Value": "Easy"}`.

In risposta alla richiesta, Amazon GameLift crea un `GameSession` oggetto che contiene la proprietà di gioco specificata. Amazon ordina GameLift quindi 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 gli ID dei giocatori

Amazon GameLift utilizza una sessione giocatore per rappresentare un giocatore connesso a una sessione di gioco. Amazon GameLift crea una sessione di gioco ogni volta che un giocatore

si connette a una sessione di gioco utilizzando un client di gioco integrato con AmazonGameLift. Quando un giocatore abbandona una partita, la sessione del giocatore termina. Amazon GameLift non riutilizza le sessioni dei giocatori.

Il seguente esempio di codice genera in modo casuale ID giocatore univoci:

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

Per ulteriori informazioni sulle sessioni dei giocatori, consulta [Visualizza i dati sulle sessioni di gioco e dei giocatori](#).

Motori di gioco e Amazon GameLift

Puoi utilizzare il GameLift servizio Amazon gestito 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 sugli GameLift SDK Amazon disponibili, sulle piattaforme di sviluppo supportate e sui sistemi operativi, consulta [Supporto allo sviluppo con Amazon GameLift](#) la pagina relativa ai server di gioco.

Oltre alle informazioni specifiche sul motore fornite in questo argomento, puoi trovare ulteriore assistenza per l'integrazione di Amazon GameLift nei tuoi server, clienti e servizi di gioco nei seguenti argomenti:

- [Roadmap per l'hosting GameLift gestito di Amazon](#)— Un flusso di lavoro in sei fasi per integrare correttamente Amazon GameLift nel tuo gioco e configurare le risorse di hosting.
- [Aggiungi Amazon GameLift al tuo server di gioco](#)— Istruzioni dettagliate sull'integrazione di Amazon GameLift in un server di gioco.
- [Aggiungi Amazon GameLift 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 utilizzando [Amazon GameLift Server SDK per C++](#). Per ottenere assistenza con l'integrazione della funzionalità desiderata nel server di gioco, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o ai tuoi servizi di gioco di interagire con il GameLift servizio Amazon, ad esempio per trovare sessioni di gioco disponibili o crearne di nuove e aggiungere giocatori ai giochi. Le funzionalità client principali sono fornite nell'[AWS SDK per C++](#). Per GameLift integrare Amazon nel tuo progetto di gioco O3DE, consulta [Aggiungi Amazon GameLift a un client e server di gioco O3DE](#) e [Aggiungi Amazon GameLift al tuo client di gioco](#)

Unreal Engine

Server di gioco

Prepara i tuoi server di gioco per l'hosting su Amazon GameLift aggiungendo [Amazon GameLift Server SDK per Unreal Engine](#) al tuo progetto e implementando le funzionalità server richieste. Per assistenza nella configurazione del plug-in Unreal Engine e nell'aggiunta del GameLift codice Amazon, consulta [Integra Amazon GameLift in un progetto Unreal Engine](#).

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o ai tuoi servizi di gioco di interagire con il GameLift servizio Amazon, ad esempio per trovare sessioni di gioco disponibili o crearne di nuove e aggiungere giocatori ai giochi. Le funzionalità client principali sono fornite nell'[AWS SDK per C++](#). Per GameLift integrare Amazon nel tuo progetto di gioco Unreal Engine, consulta [Aggiungi Amazon GameLift al tuo client di gioco](#).

Unità

Server di gioco

Prepara i tuoi server di gioco per l'hosting su Amazon GameLift aggiungendo [Amazon GameLift Server SDK per C#](#) al tuo progetto e implementando le funzionalità server richieste. Per assistenza nella configurazione con Unity e nell'aggiunta GameLift del codice Amazon, consulta [Integra Amazon GameLift in un progetto Unity](#).

Client e servizi di gioco

Consenti ai tuoi client di gioco e/o ai tuoi servizi di gioco di interagire con il GameLift servizio Amazon, 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 for .NET](#). Per GameLift integrare Amazon nel tuo progetto di gioco Unity, consulta [Aggiungi Amazon GameLift al tuo client di gioco](#).

Altri motori

Per un elenco completo degli GameLift SDK Amazon disponibili per server e client di gioco, consulta [the section called “Supporto allo sviluppo”](#).

Aggiungi Amazon GameLift a un client e server di gioco O3DE

Puoi usare 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 una struttura 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.

Amazon GameLift Gem offre le seguenti funzionalità:

GameLiftIntegrazione con Amazon

Un framework per estendere il livello di rete O3DE e consentire a Multiplayer Gem di funzionare con la soluzione server GameLift dedicata di Amazon. The Gem fornisce integrazioni sia con [l'SDK del GameLift server Amazon](#) che con il client AWS SDK (per chiamare il servizio Amazon GameLift stesso).

Gestione della compilazione 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.

Configurazione Amazon GameLift Gem

Segui le procedure in questa sezione per configurare Amazon GameLift Gem in O3DE.

Prerequisiti

- Configura il tuo AWS account per AmazonGameLift. Per ulteriori informazioni, consulta [Configura un Account AWS](#).

- Configura AWS le credenziali per O3DE. Per ulteriori informazioni, vedere [Configurazione delle AWS credenziali](#).
- Configura AWS CLI e AWS CDK. Per ulteriori informazioni, [AWS Command Line Interface](#) e [AWS Cloud Development Kit \(AWS CDK\)](#).

Attiva Amazon GameLift Gem e le sue dipendenze

1. Apri il Project Manager.
2. Apri il menu sotto il tuo progetto e scegli Modifica impostazione progetto... .
3. Scegli Configura gemme.
4. Attiva Amazon GameLift Gem e le seguenti gemme dipendenti:
 - [AWSCore Gem](#): fornisce il framework da utilizzare Servizi AWS in O3DE.
 - [Multiplayer Gem](#): fornisce funzionalità multiplayer estendendo il framework di rete.

Includi la libreria statica Amazon GameLift Gem

1. Includi `Gem::AWSGameLift.Server.Static` l'annuncio `BUILD_DEPENDENCIES` per il tuo obiettivo del server di progetto.

```
ly_add_target(  
    NAME YourProject.Server.Static STATIC  
    ...  
    BUILD DEPENDENCIES  
    PUBLIC  
    ...  
    PRIVATE  
    ...  
    Gem::AWSGameLift.Server.Static  
)
```

2. Impostato `AWSGameLiftService` su obbligatorio per il componente di sistema del server di progetto.

```
void  
YourProjectServerSystemComponent::GetRequiredServices(AZ::ComponentDescriptor::DependencyA  
required)  
{  
    ...
```

```

    required.push_back(AZ_CRC_CE("AWSGameLiftServerService"));
    ...
}

```

3. (Facoltativo) Per effettuare richieste GameLift di servizi Amazon in C++, includi `Gem::AWSGameLift.Client.Static` nel target `BUILD_DEPENDENCIES` per il tuo cliente.

```

ly_add_target(
    NAME YourProject.Client.Static STATIC
    ...
    BUILD_DEPENDENCIES
    PUBLIC
    ...
    PRIVATE
    ...
    Gem::AWSCore.Static
    Gem::AWSGameLift.Client.Static
}

```

Integra il tuo gioco e il tuo server dedicato

Gestisci le sessioni di gioco all'interno del tuo gioco e del server di gioco dedicato con l'[integrazione di gestione delle sessioni](#). Per il supporto FlexMatch, consulta [FlexMatchIntegrazione](#).

Integra Amazon GameLift in un progetto Unreal Engine

Questo argomento spiega come configurare il plug-in SDK del server Amazon GameLift C++ per Unreal Engine e integrarlo nei tuoi progetti di gioco.

Risorse aggiuntive:

- [Plugin Server SDK per il sito di download di Unreal](#)
- [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)
- [the section called “Supporto allo sviluppo”](#)

Prerequisiti

Prima di procedere, assicurati di aver esaminato i seguenti prerequisiti:

Prerequisiti

- Un computer in grado di eseguire Unreal Engine. Per ulteriori informazioni sui requisiti di Unreal Engine, consulta la documentazione sulle specifiche [hardware e software](#) di Unreal Engine.
- Microsoft Visual Studio 2019 o versione successiva.
- CMake versione 3.1 o successiva.
- Python versione 3.6 o successiva.
- Un client Git disponibile su PATH.
- Un account Epic Games. Crea un account sul sito ufficiale di [Unreal Engine](#).
- Un GitHub account associato al tuo account Unreal Engine. Per ulteriori informazioni, consulta [Accesso al codice sorgente di Unreal Engine GitHub sul](#) sito Web di Unreal Engine.

Note

Amazon GameLift attualmente supporta le seguenti versioni di Unreal Engine:

- 4.22
- 4,23
- 4,24
- 4,25
- 4,26
- 4,27
- 5.1.0
- 5.1.1
- 5.2
- 5.3

Crea Unreal Engine dai sorgenti

Le versioni standard dell'editor di Unreal Engine, scaricate tramite il launcher Epic, consentono solo build di applicazioni client Unreal. Per creare un'applicazione server Unreal, devi scaricare e creare Unreal Engine dal codice sorgente, utilizzando il repository Github di Unreal Engine. Per ulteriori

informazioni, consulta il tutorial [Building Unreal Engine from Source](#) sul sito Web di documentazione di Unreal Engine.

 Note

Se non l'hai già fatto, segui le istruzioni in [Accesso al codice sorgente di Unreal Engine su GitHub](#) per collegare il tuo GitHub account al tuo account Epic Games.

Per clonare il codice sorgente di Unreal Engine nel tuo ambiente di sviluppo


1. Clona il codice sorgente di Unreal Engine nel tuo ambiente di sviluppo in un ramo a tua scelta.

```
git clone https://github.com/EpicGames/UnrealEngine.git
```

2. Dai un'occhiata al tag della versione che stai usando per sviluppare il gioco. Ad esempio, l'esempio seguente verifica la versione 5.1.1 di Unreal Engine:

```
git checkout tags/5.1.1-release -b 5.1.1-release
```

3. Passa alla cartella principale del repository locale. Quando ti trovi nella cartella principale, esegui il seguente file: `Setup.bat`.
4. Mentre sei nella cartella principale, esegui anche il file: `GenerateProjectFiles.bat`.
5. Dopo aver eseguito i file dei passaggi precedenti, viene creato un file di soluzione Unreal EngineUE5.sln,. Apri Visual Studio e nell'editor di Visual Studio apri il UE5.sln file.
6. In Visual Studio, apri il menu Visualizza e scegli l'opzione Solution Explorer. Si apre il menu contestuale del nodo del progetto Unreal. Nella finestra Solution Explorer, fai clic con il pulsante destro del mouse sul UE5.sln file (può essere elencato come sempliceUE5), quindi scegli Build per creare il progetto Unreal con il target Development Editor Win64.

 Note

Il completamento della compilazione può richiedere più di un'ora.

Una volta completata la build, sei pronto per aprire l'Unreal Development Editor e creare o importare un progetto.

Configura il tuo progetto Unreal per il plugin

Segui questi passaggi per preparare il plug-in Amazon GameLift Server SDK per Unreal Engine per i tuoi progetti di server di gioco.

Per configurare il tuo progetto per il plugin

1. Con Visual Studio aperto, vai al riquadro Solution Explorer e scegli il UE5 file per aprire il menu contestuale del progetto Unreal. Nel menu contestuale, scegli l'opzione Imposta come progetto di avvio.
2. Nella parte superiore della finestra di Visual Studio, scegli Avvia debug (freccia verde).

Questa azione avvia la tua nuova istanza di Unreal Editor creata dal codice sorgente. Per ulteriori informazioni sull'utilizzo di Unreal Editor, consulta Unreal Editor [Interface sul sito Web di documentazione di Unreal Engine](#).

3. Chiudi la finestra di Visual Studio che hai aperto, poiché Unreal Editor apre un'altra finestra di Visual Studio che contiene il progetto Unreal e il tuo progetto di gioco.
4. Nell'editor Unreal, esegui una delle seguenti operazioni:
 - Scegli un progetto Unreal esistente che desideri integrare con Amazon GameLift.
 - Crea un nuovo progetto. Per sperimentare con il GameLift plug-in Amazon per Unreal, prova a utilizzare il modello in terza persona del motore Unreal. Per ulteriori informazioni su questo modello, consulta il modello in [terza persona sul sito Web](#) di documentazione di Unreal Engine.

In alternativa, configura un nuovo progetto con le seguenti impostazioni:

- C++
- Con contenuto iniziale
- Desktop
- Un nome di progetto. Negli esempi di questo argomento, abbiamo dato un nome al nostro progettoGameLiftUnrealApp.

5. In Solution Explorer di Visual Studio, vai alla posizione del tuo progetto Unreal. Nella Source cartella Unreal, trova un file denominato *Your-application-name*.Target.cs

Ad esempio: GameLiftUnrealApp.Target.cs.

6. Crea una copia di questo file e assegna un nome alla copia:*Your-application-name*Server.Target.cs.

7. Apri il nuovo file e apporta le seguenti modifiche:

- Modificate l'constructor in modo che corrisponda al nome del file.
- Cambia il Type da `TargetType.Game` a `TargetType.Server`
- Il file finale sarà simile al seguente esempio:

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

Il tuo progetto è ora configurato per accettare il plug-in Amazon GameLift Server SDK.

Il prossimo compito è creare le librerie SDK del server C++ per Unreal in modo da poterle importare nel progetto.

Per creare le librerie SDK del server C++ per Unreal

1. Scarica il [plug-in SDK del server Amazon GameLift C++ per Unreal](#).

Note

L'inserimento dell'SDK nella directory di download predefinita può causare un errore di compilazione dovuto al superamento del limite di 260 caratteri del percorso.

Ad esempio: `C:\Users\Administrator\Downloads\GameLift-SDK-Release-06_15_2023\GameLift-Cpp-ServerSDK-5.0.4`

Ad esempio, ti consigliamo di spostare l'SDK in un'altra directory. `C:\GameLift-Cpp-ServerSDK-5.0.4`

2. Scarica e installa OpenSSL. [Per ulteriori informazioni sul download di OpenSSL, leggi la documentazione relativa alla build e all'installazione di OpenSSL di Github.](#)

Per ulteriori informazioni, leggi la documentazione di [OpenSSL Notes per le piattaforme Windows](#).

Note

La versione di OpenSSL che usi per creare l'SDK del server GameLift Amazon deve corrispondere alla versione di OpenSSL utilizzata da Unreal per impacchettare il tuo server di gioco. Puoi trovare informazioni sulla versione nella directory di installazione di Unreal. `...Engine\Source\ThirdParty\OpenSSL`

3. Una volta scaricate le librerie, create le librerie SDK del server C++ per Unreal Engine.

Nella `GameLift-Cpp-ServerSDK-<version>` directory dell'SDK scaricato, compila con il `-DBUILD_FOR_UNREAL=1` parametro e crea l'SDK del server. I seguenti esempi mostrano come compilare utilizzando `cmake`

Esegui i seguenti comandi nel tuo terminale:

```
mkdir cmake-build
cmake.exe -G "Visual Studio 17 2022" -DCMAKE_BUILD_TYPE=Release -S . -B ./cmake-build -DBUILD_FOR_UNREAL=1 -A x64
cmake.exe --build ./cmake-build --target ALL_BUILD --config Release
```

La build di Windows crea i seguenti file binari nella `out\gamelift-server-sdk\Release` cartella:

- `cmake-build\prefix\bin\aws-cpp-sdk-gamelift-server.dll`
- `cmake-build\prefix\bin\aws-cpp-sdk-gamelift-server.lib`

Copia i due file della libreria nella `ThirdParty\GameLiftServerSDK\Win64` cartella del pacchetto del plug-in Amazon GameLift Unreal Engine.

Utilizza la seguente procedura per importare il GameLift plug-in Amazon nel tuo progetto di esempio.

Importa il GameLift plug-in Amazon

1. Individua la `GameLiftServerSDK` cartella che hai estratto dal plugin nella procedura precedente.

2. Individuala Plugins nella cartella principale del progetto di gioco. (Se la cartella non esiste, creala lì.)
3. Copia la GameLiftServerSDK cartella inPlugins.

Ciò consentirà al progetto Unreal di vedere il plugin.

4. Aggiungi il plug-in Amazon GameLift Server SDK al .uproject file del gioco.

Nell'esempio, l'app viene chiamataGameLiftUnrealApp, quindi il file saràGameLiftUnrealApp.uproject.

5. Modifica il .uproject file per aggiungere il plugin al tuo progetto di gioco.

```
"Plugins": [  
  {  
    "Name": "GameLiftServerSDK",  
    "Enabled": true  
  }  
]
```

6. Assicurati che il ModuleRules gioco dipenda dal plugin. Apri il .Build.cs file e aggiungi la dipendenza Amazon GameLiftServer SDK. Questo file si trova sotto. *Your-application-name/Source//Your-application-name/*

Ad esempio, il percorso del file del tutorial è ../GameLiftUnrealApp/Source/GameLiftUnrealApp/GameLiftUnrealApp.Build.cs.

7. Aggiungi "GameLiftServerSDK" alla fine dell'elenco diPublicDependencyModuleNames.

```
using UnrealBuildTool;  
using System.Collections.Generic;  
public class GameLiftUnrealApp : ModuleRules  
{  
    public GameLiftUnrealApp(TargetInfo Target)  
    {  
        PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject",  
"Engine", "InputCore", "GameLiftServerSDK" });  
        bEnableExceptions = true;  
    }  
}
```

Il plugin dovrebbe ora funzionare per la tua applicazione. Continua con la sezione successiva per integrare le GameLift funzionalità di Amazon nel tuo gioco.

Aggiungi il codice GameLift del server Amazon al tuo progetto Unreal

Hai configurato e configurato il tuo ambiente Unreal Engine e ora puoi integrare un server di gioco con Amazon GameLift. Il codice presentato in questo argomento effettua le chiamate obbligatorie al GameLift servizio Amazon. Implementa inoltre una serie di funzioni di callback che rispondono alle richieste del servizio Amazon GameLift . 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. [Riferimento SDK GameLift del server Amazon per Unreal Engine](#)

Per inizializzare un server di gioco con Amazon GameLift, usa la seguente procedura.

Note

Il codice GameLift specifico di Amazon fornito nella sezione seguente dipende dall'uso di un flag del WITH_GAMELIFT preprocessore. Questo flag è vero solo quando sono soddisfatte entrambe queste condizioni:

- `Target.Type == TargetRules.TargetType.Server`
- I plugin hanno trovato i file binari dell'SDK GameLift del server Amazon.

Ciò garantisce che solo le build di Unreal Server richiamino l'API di backend di GameLift Amazon. Inoltre, ti consente di scrivere codice che verrà eseguito correttamente per tutti i diversi obiettivi Unreal che il tuo gioco potrebbe produrre.

Integra un server di gioco con Amazon GameLift

1. In Visual Studio, apri il `.sln` file dell'applicazione. Nel nostro esempio, il file si `GameLiftUnrealApp.sln` trova nella cartella principale.
2. Con la soluzione aperta, individuate il `Your-application-nameGameMode.h` file dell'applicazione. Esempio: `GameLiftUnrealAppGameMode.h`.
3. Modificate il file di intestazione per allinearli al codice di esempio seguente. Assicurati di sostituire "GameLiftUnrealApp" con il nome della tua applicazione.

```
#pragma once

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

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

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

public:
    AGameLiftUnrealAppGameMode();

protected:
    virtual void BeginPlay() override;

private:
    // Process Parameters needs to remain in scope for the lifetime of the app
    FProcessParameters m_params;

    void InitGameLift();
};
```

4. Aprire il *Your-application-name*GameMode.cpp file sorgente correlato. Nel nostro esempio:GameLiftUnrealAppGameMode.cpp. e modifica il codice per allinearlo al seguente codice di esempio. Assicurati di sostituire "GameLiftUnrealApp" con il nome della tua applicazione.

Questo esempio mostra come aggiungere tutti gli elementi necessari per l'integrazione con Amazon GameLift, come descritto in [Aggiungi Amazon GameLift al tuo server di gioco](#). Questo include:

- Inizializzazione di un client Amazon GameLift API.
- Implementazione di funzioni di callback per rispondere alle richieste del GameLift servizio Amazon, tra cui OnStartGameSessionOnProcessTerminate, eonHealthCheck.

- Chiamata `ProcessReady ()` con una porta designata per avvisare Amazon GameLiftservice quando è pronto per ospitare sessioni di gioco.

```
#include "GameLiftUnrealAppGameMode.h"
#include "GameLiftUnrealAppCharacter.h"

#include "UObject/ConstructorHelpers.h"

DEFINE_LOG_CATEGORY(GameServerLog);

AGameLiftUnrealAppGameMode::AGameLiftUnrealAppGameMode()
{
    // 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;
    }
}

void AGameLiftUnrealAppGameMode::BeginPlay()
{
#ifdef WITH_GAMELIFT
    InitGameLift();
#endif
}

void AGameLiftUnrealAppGameMode::InitGameLift()
{
    UE_LOG(GameServerLog, Log, TEXT("Initializing the GameLift Server"));

    //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 serverParameters;

    //AuthToken returned from the "aws gamelift get-compute-auth-token" API. Note
    this will expire and require a new call to the API after 15 minutes.
```



```
    if (FParse::Value(FCommandLine::Get(), TEXT("-authtoken="),
serverParameters.m_authToken))
    {
        UE_LOG(GameServerLog, Log, TEXT("AUTH_TOKEN: %s"),
*serverParameters.m_authToken)
    }

    //The Host/compute-name of the GameLift Anywhere instance.
    if (FParse::Value(FCommandLine::Get(), TEXT("-hostid="),
serverParameters.m_hostId))
    {
        UE_LOG(GameServerLog, Log, TEXT("HOST_ID: %s"), *serverParameters.m_hostId)
    }

    //The Anywhere Fleet ID.
    if (FParse::Value(FCommandLine::Get(), TEXT("-fleetid="),
serverParameters.m_fleetId))
    {
        UE_LOG(GameServerLog, Log, TEXT("FLEET_ID: %s"),
*serverParameters.m_fleetId)
    }

    //The WebSocket URL (GameLiftServiceSdkEndpoint).
    if (FParse::Value(FCommandLine::Get(), TEXT("-websocketurl="),
serverParameters.m_webSocketUrl))
    {
        UE_LOG(GameServerLog, Log, TEXT("WEBSOCKET_URL: %s"),
*serverParameters.m_webSocketUrl)
    }

    //The PID of the running process
    serverParameters.m_processId = FString::Printf(TEXT("%d"),
GetCurrentProcessId());
    UE_LOG(GameServerLog, Log, TEXT("PID: %s"), *serverParameters.m_processId);

    //InitSDK establishes a local connection with GameLift's agent to enable
further communication.
    //Use InitSDK(serverParameters) for a GameLift Anywhere fleet.
    //Use InitSDK() for a GameLift managed EC2 fleet.
    gameLiftSdkModule->InitSDK(serverParameters);

    //Implement callback function onStartGameSession
    //GameLift sends a game session activation request to the game server
    //and passes a game session object with 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().
auto onGameSession = [=](Aws::GameLift::Server::Model::GameSession gameSession)
{
    FString gameId = FString(gameSession.GetGameSessionId());
    UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"),
*gameSessionId);
    gameLiftSdkModule->ActivateGameSession();
};

m_params.OnStartGameSession.BindLambda(onGameSession);

//Implement callback function OnProcessTerminate
//GameLift 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 GameLift it is shutting down.
auto onProcessTerminate = [=]()
{
    UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
    gameLiftSdkModule->ProcessEnding();
};

m_params.OnTerminate.BindLambda(onProcessTerminate);

//Implement callback function OnHealthCheck
//GameLift 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 GameLift records 'false'.
//In this example, the game server always reports healthy.
auto onHealthCheck = []()
{
    UE_LOG(GameServerLog, Log, TEXT("Performing Health Check"));
    return true;
};

m_params.OnHealthCheck.BindLambda(onHealthCheck);

//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.
m_params.port = 7777;

//Here, the game server tells GameLift where to find game session log files.
//At the end of a game session, GameLift uploads everything in the specified
//location and stores it in the cloud for access later.
TArray<FString> logfiles;
logfiles.Add(TEXT("GameLift426Test/Saved/Logs/GameLift426Test.log"));
m_params.logParameters = logfiles;

//The game server calls ProcessReady() to tell GameLift it's ready to host game
sessions.
UE_LOG(GameServerLog, Log, TEXT("Calling Process Ready"));
gameLiftSdkModule->ProcessReady(m_params);
}
```

5. 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 solo il progetto nella Games cartella che corrisponde al nome della tua app. Altrimenti Visual Studio ricostruisce l'intero progetto UE5, operazione che potrebbe richiedere fino a un'ora.

6. Una volta completate entrambe le build, chiudi Visual Studio e apri il `.uproject` file del progetto per aprirlo in Unreal Editor.
7. In Unreal Editor, impacchetta la build del server del tuo gioco. Per scegliere un bersaglio, vai su Piattaforme, Windows e seleziona ***Y our-application-name Server***.
8. Per avviare il processo di creazione dell'applicazione server, vai su Piattaforme, Windows e seleziona Package Project. Una volta completata la compilazione, dovresti avere un eseguibile. Nel caso del nostro esempio, il nome del file è `GameLiftUnrealAppServer.exe`.
9. La creazione di un'applicazione server in Unreal Editor produce due eseguibili. Uno si trova nella radice della cartella build del gioco e funge da wrapper per l'effettivo eseguibile del server.

Quando crei una GameLift flotta Amazon con la build del tuo server, ti consigliamo di inserire l'eseguibile del server effettivo come percorso di avvio della configurazione di runtime. Ad esempio, nella cartella Game Build, potresti avere un `GameLiftFPS.exe` file nella directory principale e un altro in `\GameLiftFPS\Binaries\Win64\GameLiftFPSServer.exe`.

Quando crei una flotta, ti consigliamo di `C:\GameLiftFPS\Binaries\Win64\GameLiftFPSServer.exe` utilizzarla come percorso di avvio della configurazione di runtime.

- Assicurati di aprire le porte UDP necessarie sulla GameLift flotta Amazon, in modo che il server di gioco possa comunicare con i clienti di gioco. Per impostazione predefinita, Unreal Engine utilizza la porta. 7777 Per ulteriori informazioni, consulta [UpdateFleetPortSettings](#) la guida di riferimento dell'API dei GameLift servizi Amazon.
- Crea un `install.bat` file per la build del tuo gioco. Questo script di installazione viene eseguito ogni volta che la build del gioco viene distribuita su una GameLift flotta Amazon. Ecco un `install.bat` file di esempio:

```
VC_redist.x64.exe /q
UE5PrereqSetup_x64.exe /q
```

Per alcune versioni di Unreal Engine, `install.bat` dovrebbe invece essere

```
VC_redist.x64.exe /q
UEPrereqSetup_x64.exe /q
```

Note

Il percorso del `<>PrereqSetup_x64.exe` file è `Engine\Extras\Redist\en-us`.

- Ora puoi impacchettare e caricare la build del tuo gioco su Amazon GameLift.

La versione di OpenSSL inclusa nella build del gioco deve corrispondere alla versione utilizzata dal motore di gioco per creare il server di gioco. Assicurati di includere la versione OpenSSL corretta con la build del tuo server di gioco. Per il sistema operativo Windows, il formato OpenSSL è. `.dll`

Note

Package delle DLL OpenSSL nella build del tuo server di gioco. Assicurati di impacchettare la stessa versione di OpenSSL che hai usato per creare il server di gioco.

- `libssl-1_1-x64.dll`

`libcrypto-1_1-x64.dll`

Package delle tue dipendenze insieme all'eseguibile del server di gioco nella radice di un file zip. Ad esempio, le `openssl-lib` DLL dovrebbero trovarsi nella stessa directory del `.exe` file.

Passaggi successivi

Hai configurato e configurato il tuo ambiente Unreal Engine e ora puoi iniziare a GameLift integrare Amazon nel tuo gioco.

Per ulteriori informazioni sull'aggiunta di Amazon GameLift al tuo gioco, consulta quanto segue:

- [Aggiungi Amazon GameLift al tuo server di gioco](#)
- [Riferimento SDK GameLift del server Amazon per Unreal Engine](#)

Per istruzioni su come testare il gioco, consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#).

Integra Amazon GameLift in un progetto Unity

Questo argomento spiega come configurare il plug-in Amazon GameLift C# Server SDK per Unity e integrarlo nei tuoi progetti di gioco.

Risorse aggiuntive:

- [Sito di download di Amazon GameLift Server SDK](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)
- [the section called "Supporto allo sviluppo"](#)

Prerequisiti

Per utilizzare il plug-in SDK del server Amazon GameLift C# per Unity, sono necessari i seguenti componenti:

- Un ambiente di sviluppo e una versione di Unity Editor supportati dal plugin (vedi [Supporto allo sviluppo con Amazon GameLift](#)). Per informazioni sulle versioni di Unity, consulta [Requisiti di sistema per Unity](#) nella documentazione di Unity.

- Il plug-in Amazon GameLift Server SDK per il pacchetto Unity. Questo pacchetto include il server SDK 5+ per C#. Puoi scaricare il pacchetto da questo sito: [Getting Started with Amazon. GameLift](#)
- Il registro UnityNuGet con ambito di terze parti. Questo strumento gestisce DLL di terze parti. Per ulteriori informazioni, consulta il [UnityNuGetrepository Github](#).

Configurazione di UnityNuGet

Se non hai UnityNuGet configurato il tuo progetto di gioco, usa i seguenti passaggi per installare lo strumento utilizzando il gestore di pacchetti Unity. In alternativa, puoi utilizzare la NuGet CLI per scaricare manualmente le DLL. Per ulteriori informazioni, consulta il server SDK for Unity di Amazon GameLift C#. [README](#)

Da integrare UnityNuGet nel tuo progetto di gioco

1. Con il progetto aperto in Unity Editor, vai al menu principale e seleziona Modifica, Impostazioni del progetto. Dalle opzioni, scegli la sezione Package Manager e apri il gruppo Scoped Registries.
2. Scegli il pulsante + e inserisci i seguenti valori per il UnityNuGet registro con ambito:

```
Name: Unity NuGet
URL: https://unitynuget-registry.azurewebsites.net
Scope(s): org.nuget
```

3. Per gli utenti della versione Unity 2021:

Dopo la configurazione UnityNuGet, verifica la presenza di Assembly Version Validation errori nella console Unity. Questi errori si verificano se i reindirizzamenti di associazione per gli assembly con nomi forti nei NuGet pacchetti non si risolvono correttamente nei percorsi all'interno del progetto Unity. Per risolvere questo problema, configura la convalida della versione dell'assembly di Unity:

- a. In Unity Editor, vai al menu principale e seleziona Modifica, Impostazioni progetto e apri la sezione Player.
- b. Deselezionate l'opzione di convalida della versione dell'assieme.

Installa il plugin

Utilizza la seguente procedura per installare il plug-in SDK del server Amazon GameLift C# per Unity e configurare la registrazione log4net.

Per installare il plug-in

1. Con il progetto aperto in Unity Editor, vai al menu principale e seleziona Window, Package Manager.
2. Scegli il pulsante + per aggiungere un nuovo pacchetto. Scegli l'opzione Aggiungi pacchetto da tarball.
3. In Seleziona pacchetti su disco, individua il plug-in Amazon GameLift C# Server SDK per i file di download di Unity e scegli il file Amazon GameLift Server SDK. .tgz Scegli Apri per installare il plugin.

L'SDK GameLift del server Amazon utilizza il framework log4net per generare messaggi di log. Per impostazione predefinita, è configurato per inviare messaggi al terminale di un server costruito, ma Unity richiede la configurazione per aggiungere il supporto per la registrazione dei file. Puoi aggiungere questo supporto al tuo progetto importando l'esempio fornito all'interno del pacchetto Amazon GameLift Server SDK. Utilizza la seguente procedura per aggiungere l'esempio e configurare log4net:

Per configurare log4net per l'output dei file

1. Con il progetto aperto in Unity Editor, vai al menu principale e seleziona Window, Package Manager.
2. Dal menu a discesa, seleziona Pacchetti: In Project, quindi seleziona Amazon GameLift Server SDK dall'elenco dei pacchetti. Verranno visualizzati i dettagli del pacchetto.
3. Nei dettagli del pacchetto, seleziona l'opzione del gruppo Samples e premi Importa.
4. Il `log4net.config` file e `LoggingConfiguration.cs` lo script di accompagnamento eseguono automaticamente la configurazione, che ora è impostata nella cartella del `Assets/Samples` progetto.

Note

Se è necessario spostare il `log4net.config` file in un'altra cartella del progetto, è necessario aggiornare anche il percorso del file di configurazione nello script

LoggingConfiguration.cs con il nuovo percorso. Per ulteriori informazioni, consulta il [manuale log4net sulla configurazione di log4net](#).

Per istruzioni più dettagliate e linee guida per i test, consulta il file che README si trova nel download del plugin.

Configura una GameLift Anywhere flotta Amazon per i test

Puoi configurare la tua workstation di sviluppo come flotta di GameLift Anywhere hosting Amazon per testare in modo iterativo la tua integrazione con Amazon GameLift . Con questa configurazione, puoi avviare i processi del server di gioco sulla tua workstation, inviare richieste di iscrizione o matchmaking dei giocatori GameLift ad Amazon per 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

Per istruzioni sulla configurazione della workstation, consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#) i seguenti passaggi:

1. Crea una posizione personalizzata per la tua workstation.
2. Crea una GameLift Anywhere flotta Amazon con la tua nuova sede 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 Anywhere flotta. Fornisci un nome di elaborazione univoco e specifica l'indirizzo IP della tua workstation. In caso di esito positivo, 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 tuo server di gioco lo utilizza per autenticare la comunicazione con il servizio Amazon GameLift . Archivia l'autenticazione sul computer della tua workstation in modo che i processi del server di gioco in esecuzione possano accedervi.

Aggiungi il codice GameLift del server Amazon al tuo progetto Unity

Il tuo server di gioco comunica con il GameLift servizio Amazon per ricevere istruzioni e segnalare lo stato in corso. A tale scopo, aggiungi un codice per il server di gioco che utilizza l'SDK del GameLift server Amazon.

L'esempio di codice fornito illustra gli elementi di integrazione di base richiesti. Usa a `MonoBehaviour` per illustrare una semplice inizializzazione del server di gioco con Amazon. GameLift L'esempio presuppone che il server di gioco funzioni su una GameLift Anywhere flotta Amazon per i test. Include codice per:

- Inizializza un client Amazon GameLift API. L'esempio utilizza la versione di `InitSDK()` con i parametri del server per il tuo Anywhere parco macchine e le tue capacità di calcolo. 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 GameLift Anywhere flotta Amazon per i test](#)
- Implementa le funzioni di callback per rispondere alle richieste del GameLift servizio Amazon, tra cui `OnStartGameSessionOnProcessTerminate`, e. `onHealthCheck`
- Chiama `ProcessReady()` con una porta designata per avvisare il GameLift servizio Amazon quando il processo è pronto per ospitare sessioni di gioco.

Il codice presentato in questo argomento stabilisce la comunicazione con il GameLift servizio Amazon e. Implementa inoltre una serie di funzioni di callback che rispondono alle richieste di. 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. [Riferimento SDK per GameLift server Amazon per C#](#)

Questo esempio mostra come aggiungere tutti gli elementi richiesti, come descritto in [Aggiungi Amazon GameLift al tuo server di gioco](#). Include:

Per ulteriori informazioni sull'aggiunta di GameLift funzionalità Amazon, consulta questi argomenti:

- [Aggiungi Amazon GameLift al tuo server di gioco](#)
- [Riferimento SDK per GameLift server Amazon per C#](#)

```
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 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 a GameLift Anywhere fleet.
//They are not required for a GameLift managed EC2 fleet.
ServerParameters serverParameters = new ServerParameters(
    websocketUrl,
    processId,
    hostId,
    fleetId,
    authToken);

//InitSDK establishes a local connection with an Amazon GameLift agent
//to enable further communication.
var initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);
if (initSDKOutcome.Success)
{
    //Implement callback functions
    ProcessParameters processParameters = new ProcessParameters(
    //Implement OnStartGameSession callback
        (gameSession) => {
            //GameLift 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) => {
        //GameLift 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
        //GameLift 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 GameLift it is shutting
down.
        GameLiftServerAPI.ProcessEnding();
    },
    () => {
        //Implement callback function OnHealthCheck
        //GameLift 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 GameLift
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 GameLift where to find game session
log files.

```

```
        //At the end of a game session, GameLift 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 GameLift 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 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);
    }
}
```

```
}  
}
```

Risorse aggiuntive

Usa le seguenti risorse per testare il tuo server di gioco ed espandere le funzionalità:

- Configura la tua macchina di sviluppo come flotta Amazon GameLift Anywhere e usala per i test locali. Consulta [Verifica la tua integrazione personalizzata con i server](#).
- Crea il tuo server di gioco e carica la build su Amazon GameLift. Vedi [Caricare una build di server personalizzata su Amazon GameLift](#).
- Implementa la build del tuo server di gioco su una flotta EC2 GameLift gestita da Amazon. Vedi [Creare una nuova flotta Amazon GameLift](#).

Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon

Puoi utilizzare una GameLift Anywhere flotta Amazon per creare e testare in modo iterativo l'integrazione del tuo gioco con AmazonGameLift. Configura il tuo hardware come una Anywhere flotta con una connessione al GameLift servizio Amazon, quindi installa ed esegui il tuo server di gioco su di esso. Usa un'app di test per eseguire scenari come l'avvio/interruzione delle sessioni di gioco, il monitoraggio delle connessioni dei giocatori e la gestione dei backfill del matchmaking. Con una Anywhere flotta, puoi aggiornare la build del tuo server di gioco in base alle esigenze e avere piena visibilità sull'attività di hosting.

Puoi utilizzare GameLift Anywhere le flotte Amazon con giochi integrati con Amazon GameLift Server SDK versione 5 o superiore.

Argomenti

- [Sviluppo iniziale](#)
- [Esegui iterazioni sul tuo server di gioco](#)

Sviluppo iniziale

Hai sviluppato il tuo gioco e lo stai integrando con l'SDK del GameLift server Amazon. Per testare la tua integrazione, puoi caricare ogni nuova iterazione della build del tuo server di gioco su Amazon GameLift e creare una flotta. In alternativa, l'utilizzo di una Anywhere flotta con il notebook di sviluppo consente di eseguire attività di sviluppo e test iterativi in modo più efficiente.

Utilizza le seguenti procedure per creare una Anywhere flotta e avviare una sessione di gioco sul tuo laptop utilizzando la GameLift console Amazon o il AWS Command Line Interface (AWS CLI).

Console

1. Apri la [GameLiftconsole Amazon](#).
2. Nel pannello di navigazione, sotto Hosting, scegli Posizioni.
3. Scegli Crea posizione.
4. Nella finestra di dialogo Crea posizione, effettuate le seguenti operazioni:
 - a. Inserisci un nome per la sede. Questa etichetta la posizione delle tue risorse di calcolo utilizzate da Amazon per eseguire GameLift i tuoi giochi in Anywhere flotte. I nomi delle località personalizzati devono iniziare con custom-.
 - b. Seleziona Create (Crea).
5. Per creare una Anywhere flotta, procedi come segue:
 - a. Nel pannello di navigazione, sotto Hosting, scegli Flotte.
 - b. Nella pagina Fleets (Parchi istanze) scegliere Create (Crea).
 - c. Nella fase Scegli il tipo di elaborazione, scegli Ovunque, quindi scegli Avanti.
 - d. Nella fase Definisci i dettagli della flotta, definisci la tua nuova flotta. Per ulteriori informazioni, consulta [Crea una nuova GameLift flotta Amazon](#).
 - e. Nella fase Seleziona sedi, seleziona la posizione personalizzata che hai creato.
 - f. Completa i passaggi rimanenti di creazione della flotta per creare la tua Anywhere flotta.
6. Registra il tuo laptop come risorsa di elaborazione nel parco macchine che hai creato. Usa il [register-compute](#) comando (o l'operazione [RegisterComputeAPI](#)). Includi quanto `fleet-id` creato nel passaggio precedente e aggiungi un `compute-name` e quello del tuo `laptopip-address`.

```
aws gamelift register-compute \  
  --compute-name DevLaptop \  
  --fleet-id fleet-1234 \  
  --ip-address 10.1.2.3 \  
  --location custom-location-1
```

Output di esempio:

```
Compute {
  FleetId = fleet-1234,
  ComputeName = DevLaptop,
  Status = ACTIVE,
  IpAddress = 10.1.2.3,
  GameLiftServiceSdkEndpoint = wss://12345678.execute-api.amazonaws.com/,
  Location = custom-location-1
}
```

7. Avvia una sessione di debug del tuo server di gioco.
 - a. Ottieni il token di autorizzazione per il tuo laptop del parco macchine che hai creato. Usa il [get-compute-auth-token](#) comando (o l'operazione [GetComputeAuthToken](#) API).

```
aws gamelift get-compute-auth-token \
  --fleet-id fleet-1234 \
  --compute-name DevLaptop
```

Output di esempio:

```
ComputeAuthToken {
  FleetId = fleet-1234,
  ComputeName = DevLaptop,
  AuthToken = abcdefg123,
  ExpirationTime = 1897492857.11
}
```

- b. Esegui un'istanza di debug del file eseguibile del tuo server di gioco. Per eseguire l'istanza di debug, il server di gioco deve effettuare una chiamata [InitSDK\(\)](#). Quando il processo è pronto per ospitare una sessione di gioco, il server di gioco chiama [ProcessReady\(\)](#).
8. Crea una sessione di gioco per testare la tua prima integrazione con Amazon GameLift Anywhere. Usa il [create-game-session](#) comando (o l'operazione [CreateGameSession](#) API). Specifica la posizione personalizzata della flotta.

```
aws gamelift create-game-session \
  --fleet-id fleet-1234 \
  --name DebugSession \
  --maximum-player-session-count 2 \
```

```
--location custom-location-1
```

Output di esempio:

```
GameSession {  
  FleetId = fleet-1234,  
  GameSessionId = 1111-1111,  
  Name = DebugSession,  
  IpAddress = 10.1.2.3,  
  Port = 1024,  
  ...  
}
```

Amazon GameLift invia un `onStartGameSession()` messaggio al processo del tuo server registrato. Il messaggio contiene l'`GameSession` oggetto del passaggio precedente con le proprietà del gioco, i dati delle sessioni di gioco, i dati del matchmaker e altro sulla sessione di gioco.

9. Aggiungi la logica al tuo server di gioco in modo che il processo del server risponda al `onStartGameSession()` messaggio con `ActivateGameSession()`. L'operazione invia ad Amazon una conferma GameLift che il tuo server ha ricevuto e accettato il messaggio di creazione della sessione di gioco. Per ulteriori informazioni, consultare [Riferimento SDK per GameLift server Amazon](#).

Il tuo server di gioco sta ora eseguendo una sessione di gioco che puoi testare e utilizzare per l'iterazione. Per scoprire come eseguire iterazioni sul tuo server di gioco, vai alla sezione successiva.

AWS CLI

1. Crea una posizione personalizzata utilizzando il [create-location](#) comando (o l'operazione [CreateLocation](#) API). Una posizione personalizzata indica la posizione dell'hardware GameLift utilizzato da Amazon per eseguire i giochi nelle Anywhere flotte.

```
aws gamelift create-location \  
  --location-name custom-location-1
```

Output di esempio:

```
{
```



```
Location {
    LocationName = custom-location-1
}
}
```

2. Crea una Anywhere flotta con la tua posizione personalizzata utilizzando il [create-fleet](#) comando (o l'operazione [CreateFleetAPI](#)). Amazon GameLift crea la flotta nella tua regione di origine e nelle sedi personalizzate che fornisci.

```
aws gamelift create-fleet \
  --name LaptopFleet \
  --compute-type ANYWHERE \
  --locations "location=custom-location-1"
```

Output di esempio:

```
Fleet {
    Name = LaptopFleet,
    ComputeType = ANYWHERE,
    FleetId = fleet-1234,
    Status = ACTIVE
    ...
}
```

3. Registra il tuo laptop come risorsa di elaborazione nel parco macchine che hai creato. Usa il [register-compute](#) comando (o l'operazione [RegisterComputeAPI](#)). Includi `fleet-id` ciò che è stato creato nel passaggio precedente `compute-name` e aggiungi il pubblico del tuo `laptopip-address`.

```
aws gamelift register-compute \
  --compute-name DevLaptop \
  --fleet-id fleet-1234 \
  --ip-address 10.1.2.3 \
  --location custom-location-1
```

Output di esempio:

```
Compute {
    FleetId = fleet-1234,
    ComputeName = DevLaptop,
    Status = ACTIVE,
```

```
IpAddress = 10.1.2.3,  
GameLiftServiceSdkEndpoint = wss://12345678.execute-api.amazonaws.com/,  
Location = custom-location-1  
}
```

4. Avvia una sessione di debug del tuo server di gioco.
 - a. Ottieni il token di autorizzazione per il tuo laptop del parco macchine che hai creato. Usa il [get-compute-auth-token](#) comando (o l'operazione [GetComputeAuthTokenAPI](#)).

```
aws gamelift get-compute-auth-token \  
  --fleet-id fleet-1234 \  
  --compute-name DevLaptop
```

Output di esempio:

```
ComputeAuthToken {  
  FleetId = fleet-1234,  
  ComputeName = DevLaptop,  
  AuthToken = abcdefg123,  
  ExpirationTime = 1897492857.11  
}
```

- b. Esegui un'istanza di debug del file eseguibile del tuo server di gioco. Per eseguire l'istanza di debug, il server di gioco deve effettuare una chiamata `InitSDK()`. Quando il processo è pronto per ospitare una sessione di gioco, il server di gioco chiama `ProcessReady()`.
5. Crea una sessione di gioco per testare la tua prima integrazione con Amazon GameLift Anywhere. Usa il [create-game-session](#) comando (o l'operazione [CreateGameSessionAPI](#)).

```
aws gamelift create-game-session \  
  --fleet-id fleet-1234 \  
  --name DebugSession \  
  --maximum-player-session-count 2
```

Output di esempio:

```
GameSession {  
  FleetId = fleet-1234,
```

```
GameSessionId = 1111-1111,  
Name = DebugSession,  
IpAddress = 10.1.2.3,  
Port = 1024,  
...  
}
```

Amazon GameLift invia un `onStartGameSession()` messaggio al processo del tuo server registrato. Il messaggio contiene l'oggetto `GameSession` del passaggio precedente con le proprietà del gioco, i dati delle sessioni di gioco, i dati del matchmaker e altro sulla sessione di gioco.

6. Aggiungi la logica al tuo server di gioco in modo che il processo del server risponda al `onStartGameSession()` messaggio con `activateGameSession()`. L'operazione invia ad Amazon una conferma GameLift che il tuo server ha ricevuto e accettato il messaggio di creazione della sessione di gioco. Per ulteriori informazioni, consultare [Riferimento SDK per GameLift server Amazon](#).

Il tuo server di gioco sta ora eseguendo una sessione di gioco che puoi testare e utilizzare per l'iterazione. Per scoprire come eseguire iterazioni sul tuo server di gioco, vai alla sezione successiva.

Esegui iterazioni sul tuo server di gioco

In questo caso d'uso, considera uno scenario in cui hai configurato e testato il tuo server di gioco e hai trovato un bug. Con Amazon GameLift Anywhere, puoi iterare il tuo codice ed evitare la configurazione complessa dell'utilizzo di un parco macchine Amazon EC2.

1. Ripulisci quello esistente `GameSession`, se possibile. Se il server di gioco si blocca o non chiama `processEnding()`, Amazon lo GameLift pulisce `GameSession` dopo che il server di gioco smette di inviare i controlli di integrità.
2. Apporta le modifiche al codice sul tuo server di gioco, compila e preparati per il prossimo test.
3. La Anywhere flotta precedente è ancora attiva e il notebook è ancora registrato come risorsa di elaborazione nel parco veicoli. Per ricominciare a testare, create una nuova istanza di debug.
 - a. Recupera il token di autorizzazione per il tuo laptop nel parco veicoli che hai creato. Usa il [get-compute-auth-token](#) comando (o l'operazione [GetComputeAuthTokenAPI](#)).

```
aws gamelift get-compute-auth-token \  
  --fleet-id fleet-1234 \  
  --compute-name DevLaptop
```

Output di esempio:

```
ComputeAuthToken {  
  FleetId = fleet-1234,  
  ComputeName = DevLaptop,  
  AuthToken = hijklmnop456,  
  ExpirationTime = 1897492857.11  
}
```

- b. Esegui un'istanza di debug del file eseguibile del tuo server di gioco. Per eseguire l'istanza di debug, il server di gioco deve effettuare una chiamata `InitSDK()`. Quando il processo è pronto per ospitare una sessione di gioco, il server di gioco chiama `ProcessReady()`.
4. La tua flotta dispone ora di un processo server disponibile. Crea la tua sessione di gioco ed esegui i prossimi test. Usa il [create-game-session](#) comando (o l'operazione [CreateGameSessionAPI](#)).

```
aws gamelift create-game-session \  
  --fleet-id fleet-1234 \  
  --name SecondDebugSession \  
  --maximum-player-session-count 2
```

Amazon GameLift invia un `onStartGameSession()` messaggio al processo del tuo server registrato. Il messaggio contiene l'`GameSession` oggetto del passaggio precedente con le proprietà del gioco, i dati della sessione di gioco, i dati del matchmaker e altro sulla sessione di gioco.

5. Aggiungi la logica al tuo server di gioco in modo che il processo del server risponda al `onStartGameSession()` messaggio con `ActivateGameSession()`. L'operazione invia ad Amazon una conferma GameLift che il tuo server ha ricevuto e accettato il messaggio di creazione della sessione di gioco. Per ulteriori informazioni, consultare [Riferimento SDK per GameLift server Amazon](#).

Dopo aver completato il test del server di gioco, puoi continuare a utilizzare Amazon GameLift per la gestione della tua flotta e dei server di gioco. Per ulteriori informazioni, consulta [Crea una GameLift Anywhere flotta Amazon](#).

Testa la tua integrazione con Amazon GameLift Local

Note

Utilizza questa procedura di test se utilizzi una versione di Amazon GameLift Server SDK 4.x o precedente. Il pacchetto SDK del tuo server include una versione compatibile di Amazon GameLift Local. Se utilizzi la versione 5.x del server SDK, consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#) i test locali con una flotta Amazon GameLiftAnywhere.

Usa Amazon GameLift Local per eseguire una versione limitata del GameLift servizio Amazon gestito su un dispositivo locale e testare l'integrazione del gioco con esso. Questo strumento è utile quando si esegue lo sviluppo iterativo sull'integrazione dei giochi. L'alternativa, ossia caricare ogni nuova build su Amazon GameLift e configurare una flotta per ospitare il gioco, può richiedere 30 minuti o più ogni volta.

Con Amazon GameLift Local, puoi verificare quanto segue:

- Il tuo server di gioco è correttamente integrato con il Server SDK e comunica correttamente con il GameLift servizio Amazon per avviare nuove sessioni di gioco, accettare nuovi giocatori e segnalare lo stato di salute e lo stato.
- Il tuo client di gioco è correttamente integrato con l'AWSSDK per Amazon GameLift ed è in grado di recuperare informazioni sulle sessioni di gioco esistenti, avviare nuove sessioni di gioco, unire giocatori ai giochi e connettersi alla sessione di gioco.

Amazon GameLift Local è uno strumento da riga di comando che avvia una versione autonoma del servizio Amazon gestito. GameLift Amazon GameLift Local fornisce anche un registro degli eventi in corso di inizializzazione dei processi del server, controlli di integrità e chiamate e risposte API. Amazon GameLift Local riconosce un sottoinsieme delle azioni AWS SDK per Amazon. GameLift È possibile effettuare chiamate dalla AWS CLI o dal client di gioco. Tutte le azioni API vengono eseguite localmente proprio come nel servizio GameLift web Amazon.

Ogni processo del server deve ospitare solo una singola sessione di gioco. La sessione di gioco è l'eseguibile che usi per connetterti ad Amazon GameLift Local. Una volta completata la sessione di gioco, è necessario chiamare `GameLiftServerSDK::ProcessEnding` e quindi uscire dal processo. Quando esegui i test localmente con Amazon GameLift Local, puoi avviare più processi server. Ogni processo si conetterà ad Amazon GameLift Local. 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 terminare. È quindi necessario avviare manualmente un altro processo del server.

Amazon GameLift local supporta le seguenti API:

- `CreateGameSession`
- `CreatePlayerSession`
- `CreatePlayerSessions`
- `DescribeGameSessions`
- `DescribePlayerSessions`

Configurazione GameLift locale di Amazon

Amazon GameLift Local viene fornito come `.jar` file eseguibile fornito in bundle con [Server SDK](#). Può essere eseguito su Windows o Linux e utilizzato con qualsiasi linguaggio GameLift supportato da Amazon.

Prima di eseguire Local, è necessaria l'installazione di quanto segue.

- Una build della versione Amazon GameLift Server SDK da 3.1.5 a 4.x.
- Java 8

Testa un server di gioco

Se desideri testare solo il tuo server di gioco, puoi utilizzarlo AWS CLI per simulare le chiamate dei client di gioco al servizio Amazon GameLift Local. 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 Amazon GameLift Server SDK.
- Come parte del processo di avvio, il server di gioco notifica ad Amazon GameLift che il server è pronto per ospitare sessioni di gioco.
- Il server di gioco invia lo stato di salute ad Amazon GameLift ogni minuto durante l'esecuzione.

- Il server di gioco risponde alle richieste di avvio di una nuova sessione di gioco.

1. Avvia Amazon GameLift Local.

Aprire una finestra del prompt dei comandi, accedere alla directory contenente 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.

Avvia localmente il tuo server di gioco GameLift integrato in Amazon. Non è necessario modificare l'endpoint per il server di gioco.

Nella finestra del prompt dei comandi Local, i messaggi di registro indicano che il tuo server di gioco è connesso al servizio Amazon GameLift Local. Ciò significa che il tuo server di gioco ha inizializzato correttamente l'Amazon GameLift Server SDK (`conInitSDK()`). 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 funzione, Amazon GameLift registra ogni rapporto sullo stato di salute dal 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
```

Potenziati messaggi di errore e avviso sono i seguenti:

- <process ID>Errore: "ProcessReady non ho trovato un processo con PiD:! È stato invocato initSDK ()?»
- <process ID>Avviso: «Lo stato del processo esiste già per il processo con PiD:! Viene ProcessReady (...) invocato più di una volta?»

3. Avviare 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 impostazione AWS CLI predefinita, utilizza l'endpoint del servizio GameLift web Amazon. 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 nella Guida ai [AWS CLI comandi](#).

4. Creare una sessione di gioco.

Con il 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 Local, i messaggi di registro indicano che Amazon GameLift Local ha inviato una `onStartGameSession` richiamata al tuo 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: gameId:
    "arn:aws:gamelift:local::gamesession/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-abcdef12-3456-7890-abcd-
ef1234567890"
```

Nella AWS CLI finestra, Amazon GameLift 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 cambia in `Attivo` quando il server di gioco viene richiamato `ActivateGameSession`. Se si desidera visualizzare lo stato modificato, utilizzare AWS CLI per richiamare `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
  }
}
```

```
}
```

Testa un server di gioco e un client

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 ad Amazon GameLift Local. È possibile verificare le seguenti azioni:

- Il client di gioco sta effettuando correttamente le richieste AWS SDK al servizio Amazon GameLift Local, tra cui la creazione di sessioni di gioco, il recupero di informazioni sulle sessioni di gioco esistenti e la creazione di sessioni per i giocatori.
- 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. Avvia Amazon GameLift Local.

Aprire una finestra del prompt dei comandi, accedere alla directory contenente 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.

Avvia localmente il tuo server di gioco GameLift integrato in Amazon. Per ulteriori informazioni sui log dei messaggi, consultare [Testa un server di gioco](#).

3. Configurare il client di gioco per Local e avviarlo.

Per utilizzare il tuo client di gioco con il servizio Amazon GameLift Local, devi apportare le seguenti modifiche alla configurazione del client di gioco, come descritto in [Configura Amazon GameLift 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`.
- Impostare le credenziali AWS. 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 è stato inizializzato `GameLiftClient` ed è stato comunicato correttamente con il servizio `AmazonGameLift`.

4. Prova le chiamate dei client di gioco al GameLift servizio Amazon.

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 vengono visualizzati quando Amazon GameLift Local richiede al server di gioco di avviare una sessione di gioco (`onStartGameSessioncallback`) e hanno 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 si utilizza AWS CLI per richiamare `DescribePlayerSessions()`, lo stato della sessione giocatore deve cambiare da `Reserved` (Prenotato) ad `Active` (Attivo).

6. Verifica che il tuo server di gioco stia segnalando lo stato del gioco e del giocatore al GameLift servizio Amazon.

Affinché Amazon possa GameLift gestire la domanda dei giocatori e riportare correttamente le metriche, il tuo server di gioco deve segnalare diversi stati ad Amazon. GameLift Verificare che Local registri gli eventi correlati alle operazioni seguenti. È possibile anche utilizzare la AWS CLI per tenere traccia delle modifiche di stato.

- Il giocatore si disconnette da una sessione di gioco: i messaggi di registro di Amazon GameLift Local dovrebbero mostrare che il tuo server di gioco chiama `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: i messaggi di registro di Amazon GameLift Local dovrebbero mostrare che il tuo server di gioco chiama `TerminateGameSession()`.

Note

La guida precedente consisteva nel chiamare al `TerminateGameSession()` termine di una sessione di gioco. Questo metodo è obsoleto con Amazon GameLift Server SDK v4.0.1. Consultare [Termina una sessione di gioco](#).

- Il processo del server è terminato: i messaggi di registro di Amazon GameLift Local dovrebbero mostrare che il server di gioco chiama `ProcessEnding()`. Le chiamate AWS CLI a `DescribeGameSessions()` riflettono una modifica dello stato da `Active` a `Terminated` (o `Terminating`).

Variazioni con lingua locale

Quando usi Amazon GameLift Local, tieni presente quanto segue:

- A differenza del servizio GameLift web Amazon, Local non monitora lo stato di salute di un server e non avvia la `onProcessTerminate` richiamata. Local interrompe semplicemente la registrazione dei rapporti sullo stato per il server di gioco.
- Per le chiamate all'SDK AWS, gli ID del parco istanze non vengono convalidati e possono essere qualsiasi valore di stringa che soddisfi i requisiti dei parametri (`^fleet-\S+`).
- Gli ID delle sessioni di gioco creati 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
```

Integrazione di giochi con Amazon GameLift Realtime Servers

Questo argomento fornisce una panoramica della soluzione gestita di Amazon GameLift with Realtime Servers. La panoramica spiega quando questa soluzione è adatta al tuo gioco e in che modo Realtime Servers supporta i giochi multiplayer.

Per una tabella di marcia completa sulla messa in funzione del gioco, consulta [Roadmap per l'hosting GameLift gestito di Amazon](#).

Tip

Per provare l'hosting di server di GameLift gioco Amazon, consulta [Guida introduttiva ad Amazon GameLift](#).

Cosa sono i server in tempo reale?

I server in tempo reale sono server di ready-to-go gioco leggeri che Amazon GameLift mette a tua disposizione per i tuoi giochi multiplayer. I server in tempo reale rimuovono il processo di sviluppo, test e distribuzione di un server di gioco personalizzato. Questa soluzione può aiutarti a ridurre al minimo il tempo e lo sforzo necessari per completare il gioco.

Caratteristiche principali

- Stack di rete completo per l'interazione tra client di gioco e server
- Funzionalità di base del server di gioco

- Logica del server personalizzabile
- Aggiornamenti in tempo reale alle configurazioni in tempo reale e alla logica del server
- FlexMatchmaking
- Controllo flessibile delle risorse di hosting

Configura i server in tempo reale creando una flotta e fornendo uno script di configurazione. Per ulteriori informazioni sulla creazione di server in tempo reale e su come preparare il client di gioco, consulta [Prepara il tuo server in tempo reale](#).

Come Realtime Servers gestisce le sessioni di gioco

Puoi aggiungere una logica personalizzata per la gestione delle sessioni di gioco inserendola nello script Realtime. È possibile scrivere codice per accedere a oggetti specifici del server, aggiungere logica basata su eventi utilizzando callback o aggiungere logica basata su scenari non legati a eventi.

Come interagiscono client e server in tempo reale

Durante una sessione di gioco, i client di gioco interagiscono inviando messaggi al server Realtime tramite un servizio di backend. Il servizio di backend trasmette quindi i messaggi tra i client di gioco per scambiare attività, stato del gioco e dati di gioco pertinenti.

Inoltre, puoi personalizzare il modo in cui client e server interagiscono aggiungendo la logica di gioco allo script Realtime. Con una logica di gioco personalizzata, un server in tempo reale potrebbe implementare callback per avviare risposte basate sugli eventi.

Protocollo di comunicazione

I server in tempo reale e i client di gioco connessi comunicano attraverso due canali: una connessione TCP per una distribuzione 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 invia messaggi utilizzando TCP come riserva.

Contenuto del messaggio

Il contenuto del messaggio è composto da due elementi: un codice dell'operazione (opCode) e un payload opzionale. Il codice operativo 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 contenuti nei messaggi che riceve.

Gruppi di giocatori

Realtime Servers offre funzionalità per gestire gruppi di giocatori. Per impostazione predefinita, Amazon GameLift colloca tutti i giocatori che si connettono a un gioco 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.

Server in tempo reale con certificati TLS

Con Realtime Servers, 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 server in tempo reale, il server risponde automaticamente con il certificato TLS, che il client convalida. Amazon GameLift gestisce la crittografia utilizzando TLS per la comunicazione TCP (WebSockets) e DTLS per il traffico UDP.

Personalizzazione di un server in tempo reale

Un server in tempo reale funziona come server di inoltro stateless. Il server in tempo reale trasmette pacchetti di messaggi e dati di gioco tra i client di gioco connessi al gioco. Tuttavia, il server Realtime 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 la funzionalità degli script in tempo reale. Con la logica di gioco, ad esempio, puoi creare un gioco di stato con una visione autorevole dello stato del gioco da parte del server.

Amazon GameLift definisce una serie di callback lato server per gli script in tempo reale. Implementa queste chiamate per aggiungere funzionalità basate su eventi al server. Ad esempio, sono possibili le seguenti operazioni:

- 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 inviare messaggi da un determinato giocatore o a un giocatore bersaglio, oppure esegui un'elaborazione aggiuntiva 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 utilizza i dati.

Implementazione e aggiornamento di server in tempo reale

Uno dei vantaggi principali di Realtime Servers è la possibilità di aggiornare gli script in qualsiasi momento. Quando aggiorni uno script, Amazon GameLift distribuisce la nuova versione a tutte le risorse di hosting in pochi minuti. Dopo che Amazon avrà GameLift distribuito il nuovo script, tutte le nuove sessioni di gioco create successivamente utilizzeranno la nuova versione dello script. (Le sessioni di gioco esistenti continueranno a utilizzare la versione originale.)

Inizia a integrare il tuo gioco con Realtime Servers:

- [Integrazione di un client di gioco per server in tempo reale](#)
- [Creazione di uno script in tempo reale](#)

Integrazione di un client di gioco per server in tempo reale

Questo argomento descrive come preparare il client di gioco a partecipare e partecipare a sessioni di gioco GameLift ospitate da Amazon.

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.
- Consenti al tuo client di gioco di partecipare a una sessione di gioco ospitata su un server in tempo reale e di scambiare messaggi.

Trova o crea sessioni di gioco e sessioni per i 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 best practice, crea un servizio di backend e utilizzalo per inviare richieste dirette al GameLift servizio Amazon quando vengono attivate da un'azione del client di gioco. Il servizio di backend trasmette quindi le risposte pertinenti al client di gioco.

1. Aggiungi l'AWSSDK al tuo client di gioco, inizializza un GameLift client Amazon e configuralo per utilizzare le risorse di hosting delle tue flotte e delle tue code. L'AWSSDK è disponibile in diverse lingue; consulta gli GameLift SDK [Per servizi client personalizzati](#) Amazon.
2. Aggiungi GameLift funzionalità al tuo servizio di backend. Per istruzioni più dettagliate, vedi [Aggiungi Amazon GameLift al tuo client di gioco](#) e [Aggiungere il 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 capacità GameLift di inserire 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 tuo 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 backend riceve gli oggetti della sessione di gioco e della sessione del giocatore in risposta alle richieste al GameLift servizio Amazon. 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.

Connettiti ai giochi su server in tempo reale

Consenti al tuo client di gioco di connettersi direttamente a una sessione di gioco ospitata su un server in tempo reale e di scambiare messaggi con il server e con altri giocatori.

1. Scarica Realtime Client SDK, crealo e aggiungilo al tuo progetto 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 specifichi il tipo di connessione client/server 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 [Riferimento all'API client \(C#\) di Realtime Servers](#).

- 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 [Riferimento API client \(C#\) di Realtime Servers: 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 in tempo reale di base (C#)

Questo esempio illustra un'integrazione di base del client di gioco con Realtime Client SDK (C#). Come illustrato, l'esempio inizializza un oggetto client in tempo reale, configura i gestori di eventi e implementa i callback lato client, si connette a un server in tempo reale, 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 GameLift Realtime client SDK
     *
     * 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 GameLift 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 GameLift
    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 script in tempo reale

Per utilizzare Realtime Servers per il tuo gioco, devi fornire uno script (sotto forma di JavaScript codice) per configurare e, facoltativamente, personalizzare una flotta di server in tempo reale. Questo argomento descrive i passaggi chiave per la creazione di uno script in tempo reale. Una volta che lo script è pronto, caricalo sul GameLift servizio Amazon e usalo per creare una flotta (vedi [Carica uno script Realtime Servers su Amazon GameLift](#)).

Per preparare uno script da utilizzare con Realtime Servers, aggiungi la seguente funzionalità al tuo script Realtime.

Gestisci il ciclo di vita delle sessioni di gioco (obbligatorio)

Come minimo, uno script in tempo reale deve includere la `Init()` funzione, che prepara il server Realtime ad 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.

Alla funzione di `Init()` callback, quando viene chiamata, viene passato un oggetto di sessione in tempo reale, che contiene un'interfaccia per il server Realtime. Per ulteriori informazioni su questa interfaccia, consulta [Interfaccia Realtime Servers](#).

Per terminare correttamente una sessione di gioco, lo script deve anche richiamare la funzione del server `Realtime.session.processEnding`. 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.

I server in tempo reale con la configurazione più semplice (inizializzazione e terminazione dei processi del server) agiscono essenzialmente come server di inoltro senza stato. Il server in tempo reale trasmette messaggi e dati di gioco tra i client di gioco connessi al gioco, ma non intraprende alcuna azione indipendente per elaborare i dati o eseguire la 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)

Facoltativamente, puoi aggiungere la logica di gioco al tuo script in tempo reale. Ad esempio, è possibile eseguire una o tutte le seguenti operazioni. Il codice di esempio di script fornisce illustrazione. Consultare [Riferimento allo script Amazon GameLift Realtime Servers](#).

- 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 server in tempo reale](#).
- 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 `onMessage` e analizza il contenuto del messaggio tramite l'interfaccia `gameMessage` (consulta [gameMessage.opcode](#)).
- Abilita la logica di gioco per accedere alle tue 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 dell'istanza su cui è in esecuzione. Per informazioni dettagliate, consultare [Ottieni dati sulla flotta per un'GameLiftistanza Amazon](#).

Esempio di script Realtime Servers

Questo esempio illustra uno script di base necessario per implementare Realtime Server più una 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 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 GameLift
// 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  
};
```

Integrazione di giochi con il GameLift plug-in Amazon per Unity

Gli argomenti di questa sezione descrivono il GameLift plug-in Amazon per Unity e come utilizzarlo per preparare il tuo progetto di gioco multiplayer per l'hosting con Amazon GameLift. Lavora interamente nel tuo ambiente di sviluppo Unity con i flussi di lavoro guidati del plug-in per completare i requisiti di base per l'hosting con Amazon GameLift.

Amazon GameLift è un servizio completamente gestito che consente agli sviluppatori di giochi di gestire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni. Per ulteriori informazioni sull' GameLift hosting Amazon, consulta [Come GameLift funziona Amazon](#).

- [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#), versione 2.0.0, funziona con il server SDK 5.x e supporta Amazon. GameLift Anywhere
- [Guida al GameLift plug-in Amazon per Unity per server SDK 4.x](#), versione 1.0.0, funziona con server SDK 4.x o versioni precedenti. Questa versione utilizza Amazon GameLift Local per i test di integrazione.

Guida al GameLift plug-in Amazon per Unity per server SDK 5.x

Amazon GameLift fornisce strumenti per preparare i server di gioco multigiocatore in modo che funzionino con Amazon GameLift. Il GameLift plug-in Amazon per Unity semplifica l'integrazione

di Amazon GameLift nei tuoi progetti di gioco Unity, il test dell'integrazione con Amazon GameLift Anywhere e la distribuzione GameLift delle risorse Amazon per il cloud hosting.

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

Argomenti

- [Informazioni sul plugin](#)
- [Workflow del plug-in](#)
- [Installa il plugin per Unity](#)
- [Configura un profilo AWS utente](#)
- [Configura il gioco per i test locali con Amazon GameLift Anywhere](#)
- [Distribuisci il tuo gioco sull'hosting cloud con flotte EC2 gestite](#)

Informazioni sul plugin

Il plug-in per Unity offre un'esperienza iniziale semplificata per l'integrazione e l'hosting dei tuoi giochi multiplayer Unity con Amazon. GameLift Puoi sfruttare le funzionalità dei plug-in e i componenti predefiniti per far funzionare rapidamente i tuoi giochi.

Il plugin aggiunge strumenti e funzionalità all'editor Unity. Usa i flussi di lavoro guidati per GameLift integrare Amazon nel tuo progetto di gioco, testarlo localmente e poi distribuire il server di gioco sull'hosting GameLift cloud Amazon.

Utilizza le soluzioni di hosting predefinite del plug-in per distribuire il tuo gioco. Configura una flotta Amazon GameLift Anywhere con la tua workstation locale come host. Per l'hosting su cloud, scegli tra due scenari di implementazione comuni che bilanciano la latenza dei giocatori, la disponibilità delle sessioni di gioco e i costi in modi diversi. Uno scenario include un semplice FlexMatch matchmaker e un set di regole. Utilizza questi scenari per implementare una soluzione di hosting di base pronta per la produzione, quindi ottimizza e personalizza secondo necessità.

Il plugin include i seguenti componenti:

- Moduli plug-in per l'editor Unity. Una volta installato il plug-in, una nuova voce del menu principale consente di accedere alle GameLift funzionalità di Amazon.
- Librerie C# per l'API del GameLift servizio Amazon con funzionalità lato client.

- Librerie C# per l'SDK del GameLift server Amazon (versione 5.x).
- Prova i contenuti di gioco, tra cui risorse e scene, così puoi provare Amazon GameLift anche se non hai un gioco multiplayer 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.

Workflow del plug-in

I passaggi seguenti descrivono un approccio tipico all'integrazione e alla distribuzione di un progetto di gioco con il GameLift plug-in Amazon per Unity. Completi questi passaggi utilizzando l'editor di Unity e il codice del gioco.

1. Crea un profilo utente che si colleghi al tuo AWS account e fornisca le credenziali di accesso per un utente con account valido con le autorizzazioni per utilizzare Amazon. GameLift
2. Aggiungi il codice server al tuo progetto di gioco per stabilire la comunicazione tra un server di gioco in esecuzione e il GameLift servizio con Amazon.
3. Aggiungi un codice client al tuo progetto di gioco che consente ai client di gioco di inviare richieste GameLift ad Amazon per avviare o partecipare a una sessione di gioco e quindi connettersi al server di gioco.
4. Usa il flusso di lavoro Anywhere per configurare la tua workstation locale come host Anywhere per il tuo server di gioco. Avvia il server e il client di gioco localmente, connessi a una sessione di gioco e verifica l'integrazione.
5. Usa il flusso di lavoro di hosting EC2 per caricare il tuo server di gioco integrato e implementare una soluzione di hosting cloud. Quando il server di gioco è pronto, avvia il client di gioco localmente, connessi a una sessione di gioco, accedi e gioca.

Quando utilizzi il plug-in, creerai e utilizzerai AWS risorse. Queste azioni potrebbero comportare addebiti sull'AWSaccount in uso. Se sei un principianteAWS, le azioni potrebbero essere incluse nel piano [AWSgratuito](#).

Installa il plugin per Unity

Questa sezione descrive come aggiungere il plugin a un progetto Unity. Dopo l'installazione del plug-in, la funzionalità del plug-in è disponibile quando il progetto è aperto nell'editor Unity.

Prima di iniziare

Ecco cosa ti serve per utilizzare il GameLift plug-in Amazon per Unity:

- Unity per Windows 2022 LTS o Unity per macOS
- Download GameLift del plug-in Amazon per Unity. [\[Sito di download\]](#) Il download include due pacchetti:
 - GameLift Plugin Amazon autonomo per Unity
 - SDK per server Amazon GameLift C# per Unity
- Microsoft Visual Studio 2019 o versione successiva.
- Un progetto di gioco multiplayer con codice di gioco C#.
- Il registro con ambito di terze parti. UnityNuGet Questo strumento gestisce DLL di terze parti. Per ulteriori informazioni, consulta il [UnityNuGetrepository Github](#).

Aggiungi il plugin al tuo progetto di gioco

Completa le seguenti attività, utilizzando l'editor di Unity e i file di progetto del gioco.

Passaggio 1: aggiungi UnityNuGet al tuo progetto di gioco

Se non hai UnityNuGet configurato il tuo progetto di gioco, usa i seguenti passaggi per installare lo strumento utilizzando il gestore di pacchetti Unity. In alternativa, puoi utilizzare la NuGet CLI per scaricare manualmente le DLL. Per ulteriori informazioni, consulta l'[SDK per server Amazon GameLift C# for Unity. README](#)

1. Con il progetto aperto nell'editor Unity, vai al menu principale e seleziona Modifica, Impostazioni del progetto. Dalle opzioni, scegli la sezione Package Manager e apri il gruppo Scoped Registries.
2. Scegli il pulsante + e inserisci i seguenti valori per il UnityNuGet registro con ambito:

```
Name: Unity NuGet
URL: https://unitynuget-registry.azurewebsites.net
Scope(s): org.nuget
```

Per gli utenti della versione Unity 2021:

Dopo la configurazione UnityNuGet, verifica la presenza di `Assembly Version Validation` errori nella console Unity. Questi errori si verificano se i reindirizzamenti di associazione per gli assembly con nomi forti nei NuGet pacchetti non si risolvono correttamente nei percorsi all'interno del progetto Unity. Per risolvere questo problema, configura la convalida della versione dell'assembly di Unity:

1. Nell'editor Unity, vai al menu principale e seleziona Modifica, Impostazioni progetto e apri la sezione Player.
2. Deseleziona l'opzione di convalida della versione dell'assieme.

Passaggio 2: aggiungere il plug-in e i pacchetti SDK del server C#

1. Decomprimi il GameLift plug-in Amazon per il download di Unity, che contiene entrambi i pacchetti.
2. Con il progetto aperto in Unity Editor, vai al menu principale e seleziona Window, Package Manager.
3. Scegli il pulsante + per aggiungere un nuovo pacchetto. Scegli l'opzione Aggiungi pacchetto da tarball.
4. In Seleziona pacchetti su disco, individua il plug-in Amazon GameLift C# Server SDK per i file di download di Unity e scegli il `com.amazonaws.gameliftserver.sdk-<version>.tgz` file. Scegli Apri per installare il plugin.
5. In Seleziona pacchetti su disco, individua il plug-in GameLift standalone Amazon per i file di download di Unity e scegli il file `com.amazonaws.gamelift-<version>.tgz`. Scegli Apri per installare il plugin.
6. Verifica che il plugin standalone sia aggiunto al tuo progetto. Torna alla finestra dell'editor di Unity. Controlla il menu principale per il nuovo pulsante del GameLift menu Amazon.

Passaggio 3: Importa il gioco di esempio (opzionale)

Il plugin per Unity include una serie di risorse di gioco di esempio, tra cui scene, che puoi aggiungere al tuo progetto di gioco. L'importazione del gioco di esempio ti offre un percorso rapido per testare, creare e distribuire un semplice gioco multiplayer con Amazon. GameLift Il gioco di esempio è già completamente integrato con GameLift gli SDK di Amazon, quindi puoi saltare le attività di integrazione e completare le restanti attività del flusso di lavoro.

Quando usi il gioco di esempio, puoi configurare e unirti a una flotta Amazon GameLift Anywhere ospitata localmente in pochi minuti. Puoi distribuire il gioco su Amazon GameLift e partecipare a una partita live ospitata sul cloud in meno di un'ora.

Per importare il gioco di esempio:

1. Con il progetto di gioco aperto in Unity Editor, vai al GameLift menu Amazon e seleziona Sample Game, Import Sample Game.
2. Dopo aver importato i file, vai nuovamente al GameLift menu Amazon e seleziona Sample Game, Initialize Settings. Questo passaggio configura il tuo progetto per la creazione del client e del server di gioco.

Una volta completata l'installazione, vedrai due nuove scene aggiunte al progetto del gioco. Vedrai anche alcune risorse aggiuntive del progetto, inclusa una GameLiftClientSettingsrisorsa.

Per maggiori dettagli sull'interfaccia utente e sul gameplay dell'esempio, consulta il file readme del gioco di esempio.

Configura un profilo AWS utente

Dopo aver installato il plug-in, configura un profilo e collegalo a un AWS account utente valido. Puoi mantenere più profili, ma puoi avere solo un profilo attivo alla volta. Ogni volta che lavori nel plugin, seleziona un profilo da usare.

Il mantenimento di più profili ti dà la possibilità di passare da uno scenario di hosting all'altro. Ad esempio, potresti configurare profili con le stesse AWS credenziali ma AWS regioni diverse. Oppure potresti configurare profili con AWS account diversi o con utenti/set di autorizzazioni diversi.

Note

Se hai installato la AWS CLI sulla tua workstation e disponi di un profilo già configurato, il GameLift plug-in Amazon può rilevarlo e lo elencherà come profilo esistente. Il plug-in seleziona automaticamente qualsiasi profilo denominato. [default] È possibile utilizzare un profilo esistente o crearne uno nuovo.

Per configurare il tuo AWS profilo

1. Nel menu principale dell'editor Unity, scegli Amazon GameLift e seleziona Imposta profili AWS account. Questa azione apre la finestra del plugin. Apri la pagina Profili AWS utente.
2. Se il plugin rileva un profilo esistente, non ti verrà richiesto di crearne uno. Seleziona un profilo esistente o scegli Aggiungi un altro profilo per crearne uno nuovo.
3. Se il plugin non rileva un profilo esistente, ti chiede di crearne uno. Puoi creare un nuovo profilo utilizzando un account nuovo o esistente AWS.

Note

È necessario utilizzare la console di AWS gestione per creare un nuovo AWS account e creare o aggiornare un utente con il set di autorizzazioni appropriato.

Quando si configura un profilo, sono necessarie le seguenti informazioni:

- Un account AWS. Se devi creare un nuovo AWS account, segui le istruzioni per creare l'account. Vedi [Creare un AWS account](#) per maggiori dettagli.
 - Un utente dell'AWS account con le autorizzazioni per utilizzare Amazon GameLift e altri AWS servizi richiesti. Consulta [Configura un Account AWS](#) le istruzioni su come configurare un utente AWS Identity and Access Management (IAM) con GameLift autorizzazioni Amazon.
 - Credenziali per il tuo utente. AWS Questo utente necessita inoltre dell'accesso programmatico con credenziali a lungo termine. Queste credenziali sono costituite da un ID della chiave di AWS accesso e da una chiave segreta AWS. Per maggiori dettagli, consulta [Ottenere le chiavi di accesso](#).
 - Regione AWS. 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 per ridurre al minimo la latenza. Consulta l'elenco delle [AWS regioni supportate](#).
4. Quando hai selezionato o creato un profilo, controlla lo stato di bootstrap del profilo e agisci secondo necessità. Tutti i profili devono essere avviati per utilizzare la funzionalità del GameLift plug-in Amazon.

Per avviare il tuo profilo:

Il bootstrap designa un bucket Amazon S3 da utilizzare con il profilo utente selezionato. Amazon S3 è un AWS servizio fondamentale per lo storage di dati e oggetti. Il bucket utilizzato per archiviare configurazioni di progetto, elementi di creazione e altre dipendenze. I bucket non vengono condivisi tra altri profili.

Note

Il bootstrap crea nuove AWS risorse e potrebbe comportare costi.

1. Quando visualizzi i tuoi profili nella finestra del plugin Profili AWS utente, seleziona il profilo che desideri utilizzare. Se il profilo non è ancora stato avviato, viene visualizzato un messaggio di avviso.
2. Nella sezione Bootstrap il tuo profilo, seleziona un profilo dall'elenco a discesa e controlla lo stato del bootstrap. Se lo stato indica che non esiste alcun bucket, scegli il pulsante Profilo Bootstrap. Puoi impostare il nome del bucket con un nuovo nome di bucket, inserire un bucket esistente a cui hai accesso o mantenere il nome generato automaticamente.
3. Attendi che lo stato del bootstrap passi a «Attivo». Ciò può richiedere alcuni minuti. Quando lo stato è «Attivo», puoi utilizzare il profilo per lavorare con le funzionalità del plugin

Configura il gioco per i test locali con Amazon GameLift Anywhere

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

Per avviare il flusso di lavoro Amazon GameLift Anywhere:

- Nel menu principale dell'editor di Unity, scegli Amazon GameLift e seleziona Host with Anywhere. Questa azione apre la pagina del plugin per configurare il gioco con una Anywhere flotta @. 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 GameLift menu Amazon 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

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 GameLift servizio Amazon 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](https://github.com/aws/amazon-gamelift-plugin-unity) e apri il percorso. Samples~/SampleGame/Assets/Scripts/Server

3. Individua il file `GameLiftServer.cs` e copialo nella cartella `Server` del progetto di gioco. Quando crei un eseguibile sul server, usa questo file come destinazione di compilazione.

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

- Inizializza un client Amazon GameLift API. La `InitSDK()` chiamata con i parametri del server è necessaria per una flotta Amazon GameLift Anywhere. Queste impostazioni vengono impostate automaticamente per l'uso nel plug-in.
- Implementa le funzioni di callback richieste per rispondere alle richieste del GameLift servizio Amazon, tra cui `OnStartGameSessionOnProcessTerminate`, e `onHealthCheck`
- Chiamate `ProcessReady()` con una porta designata per notificare al GameLift servizio Amazon 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 al tuo server di gioco](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)

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. Usa le API del GameLift servizio Amazon (nell'AWSSDK) per ottenere informazioni sulle sessioni 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 GameLift servizio Amazon. Quando sviluppi il tuo gioco per il cloud hosting o se prevedi di utilizzare le flotte 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

Per GameLift integrare Amazon nel tuo codice cliente, usa 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 riportate in `Aggiungi Amazon GameLift al tuo client 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'AWSSDK per. GameLift Il client di gioco utilizza queste credenziali per autenticare le richieste al servizio Amazon GameLift .

Note

Il profilo delle credenziali viene generato dal plug-in 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'AWSaccount 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 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 Unreal.

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 verifica di potersi connettere 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 plug-in utilizza questa risorsa per memorizzare determinate informazioni di cui il client di gioco ha bisogno per connettersi al GameLift servizio Amazon.
 - a. Se non hai importato e inizializzato il gioco di esempio, crea una nuova `GameLiftClientSettings` risorsa. Nel menu principale dell'editor di Unity, scegli Risorse, Crea GameLift, Impostazioni client. Se crei più copie del `GameLiftClientSettings` tuo progetto, il plugin 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 stai usando 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-token](#) con 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]
```

Distribuisci il tuo gioco sull'hosting cloud con flotte EC2 gestite

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 Aggiungi il codice di gioco client e server per le GameLift funzionalità Amazon, quindi carica la build del server sul GameLift servizio Amazon 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 flusso di lavoro Amazon EC2 GameLift gestito da Amazon:

- Nel menu principale dell'editor Unity, scegli Amazon GameLift e 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'AWSaccount 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 GameLift menu Amazon 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

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 GameLift servizio Amazon 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](https://github.com/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 C# (versione 5):

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

- Chiamate `ProcessReady()` con una porta designata per notificare al GameLift servizio Amazon 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 al tuo server di gioco](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)

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 GameLift servizio Amazon, anziché effettuare le chiamate direttamente dal client di gioco.

Nel flusso di lavoro dei plug-in per l'hosting su una flotta EC2 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 ID giocatore unici 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 [GameLiftCoreApi](#) classe 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 al tuo client di gioco Unity](#).

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:
 - AWSfleet (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 alle API.
 - WebACL 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.

- AWSCode 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 alle API.
- WebACL 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 la tua flotta EC2 gestita. Amazon GameLift utilizza questo nome (insieme all'ID della flotta) per elencare le risorse nella AWS console.
- Nome build: fornisci un nome significativo per la build del tuo server. AWS usa questo nome per fare riferimento alla copia della build del tuo server caricata su Amazon GameLift e utilizzata per le distribuzioni.
- Parametri di avvio: inserisci le istruzioni opzionali da eseguire all'avvio del server eseguibile su un'istanza gestita della flotta EC2. 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, distribuisce il gioco su una soluzione di hosting cloud in base allo scenario di implementazione scelto. Questo processo può richiedere fino a 40 minuti per la AWS 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'istanza AWS EC2. Almeno un processo 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 plug-in utilizza questa risorsa per memorizzare determinate informazioni di cui il client di gioco ha bisogno per connettersi al GameLift servizio Amazon.
 - a. Se non hai importato e inizializzato il gioco di esempio, crea una nuova `GameLiftClientSettings` risorsa. Nel menu principale dell'editor di Unity, scegli Risorse, Crea GameLift, Impostazioni client. Se crei più copie del `GameLiftClientSettings` tuo progetto, il plugin lo rileva automaticamente e ti notifica quale risorsa verrà aggiornata dal plug-in.
 - b. In Launch Game, scegli Configura client: applica impostazioni EC2 gestite. Questa azione aggiorna le impostazioni del client di gioco per utilizzare la flotta EC2 gestita che hai appena implementato.
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.

Guida al GameLift plug-in Amazon per Unity per server SDK 4.x

Note

Questo argomento fornisce informazioni per una versione precedente del GameLift plug-in Amazon per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'Amazon GameLift server SDK 4.x o precedente. Per la documentazione sulla versione più recente del plug-in, che utilizza il server SDK 5.x e supporta Amazon GameLift Anywhere, consulta. [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#)

Amazon GameLift fornisce strumenti per preparare i tuoi server di gioco multiplayer per l'esecuzione su Amazon GameLift. Il GameLift plug-in Amazon per Unity semplifica l'integrazione di Amazon GameLift nei tuoi progetti di gioco Unity e la distribuzione GameLift delle risorse Amazon per l'hosting su cloud. Usa il plug-in per Unity per accedere alle GameLift API di Amazon e distribuire AWS CloudFormation modelli per scenari di gioco comuni.

Dopo aver configurato il plug-in, puoi provare l'[esempio di Amazon GameLift Unity](#) su GitHub.

Argomenti

- [Integra Amazon GameLift con un progetto di server di gioco Unity](#)
- [Integra Amazon GameLift con un progetto client di gioco Unity](#)
- [Installa e configura il plugin](#)
- [Prova il gioco localmente](#)
- [Implementa uno scenario](#)
- [Integra giochi con Amazon GameLift in Unity](#)
- [Importa ed esegui un gioco di esempio](#)

Integra Amazon GameLift con un progetto di server di gioco Unity

Note

Questo argomento fornisce informazioni per una versione precedente del GameLift plug-in Amazon per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'Amazon GameLift server SDK 4.x o precedente. Per la documentazione sulla versione più recente del plug-in, che utilizza il server SDK 5.x e supporta Amazon GameLift Anywhere, consulta [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#)

Questo argomento ti aiuta a preparare il tuo server di gioco personalizzato per l'hosting su Amazon GameLift. Il server di gioco deve essere in grado GameLift di notificare ad Amazon 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 al tuo server di gioco](#).

Prerequisiti

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

- [Configura un ruolo di servizio IAM per Amazon GameLift](#)
- [Installa il plugin per Unity](#)

Configura un nuovo processo del server

Note

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

Imposta la comunicazione con Amazon GameLift 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 GameLift servizio Amazon, ad esempio, `OnGameSession()`, `OnGameSessionUpdate()`, `OnProcessTerminate()`, `OnHealthCheck()`. Amazon GameLift potrebbe impiegare alcuni minuti per effettuare una richiamata.
3. Amazon GameLift aggiorna lo stato del processo del server su `ACTIVE`.
4. Amazon GameLift chiama `onHealthCheck` periodicamente.

Il seguente esempio di codice mostra come configurare un semplice processo server con Amazon GameLift.

```
//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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

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

1. Implementare la funzione di richiamo `onStartGameSession`. Amazon GameLift richiama 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, consulta [Riferimento Amazon GameLift Server SDK \(C#\): azioni](#).

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

```
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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Avvisa Amazon GameLift 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 `onProcessTerminate` per ricevere richieste da Amazon GameLift e chiamare `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

Note

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

Dopo aver integrato il tuo server di gioco con Amazon GameLift, carica i file di build su una flotta in modo che Amazon GameLift possa distribuirli per l'hosting di giochi. Per ulteriori informazioni su come caricare il tuo server su Amazon GameLift, consulta [Carica un server personalizzato su Amazon GameLift](#).

Integra Amazon GameLift con un progetto client di gioco Unity

Note

Questo argomento fornisce informazioni per una versione precedente del GameLift plug-in Amazon per Unity. La versione 1.0.0 (rilasciata nel 2021) utilizza l'Amazon GameLift server SDK 4.x o precedente. Per la documentazione sulla versione più recente del plug-in, che utilizza il server SDK 5.x e supporta Amazon GameLift Anywhere, consulta. [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#)

Questo argomento ti aiuta a configurare un client di gioco per connetterti alle sessioni di gioco GameLift ospitate da Amazon tramite un servizio di backend. Usa le GameLift API di Amazon 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 GameLift servizio Amazon. Un servizio di backend gestisce tutte le comunicazioni del client di gioco con il GameLift servizio. Per ulteriori informazioni sui servizi di backend, consulta. [Progetta il tuo servizio client di gioco](#)

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

- Personalizza l'autenticazione per i tuoi giocatori.
- Richiedi informazioni sulle sessioni di gioco attive tramite il GameLift servizio Amazon.
- 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

- [Prerequisiti](#)
- [Inizializza un client di gioco](#)
- [Crea una sessione di gioco su una flotta specifica](#)
- [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 GameLift client Amazon, completa le seguenti attività:

- [Configura un Account AWS](#)
- [Installa il plugin per Unity](#)
- [Integra Amazon GameLift con un progetto di server di gioco Unity](#)
- [Configurazione delle GameLift flotte Amazon](#)

Inizializza un client di gioco

Note

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

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

1. `InizializzaAmazonGameLiftClient`. Chiama `AmazonGameLiftClient` con una configurazione client predefinita o una configurazione personalizzata. Per ulteriori informazioni su come configurare un client, vedere [Configura Amazon GameLift 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 gli ID dei giocatori](#).

I seguenti esempi mostrano come configurare un GameLift client Amazon.

```
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 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 GameLift 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza 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 GameLift che Amazon ha creato la nuova sessione di gioco e restituito una `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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Dopo GameLift che Amazon ha creato la nuova sessione di gioco e 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 playerId, string playerSessionId)
{
    //Call ConnectPlayer with player ID and player session ID.
    return server.ConnectPlayer(playerId, playerSessionId);
}
```



```
}
```

Rimuovere un giocatore da una sessione di gioco

Note

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

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

1. Notifica al GameLift servizio Amazon 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 modifica lo slot del giocatore in modo che sia disponibile, che Amazon GameLift può assegnare a un nuovo giocatore.

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

```
public void DisconnectPlayer(int playerId)
{
    //Receive the player session ID.
    string playerId = 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

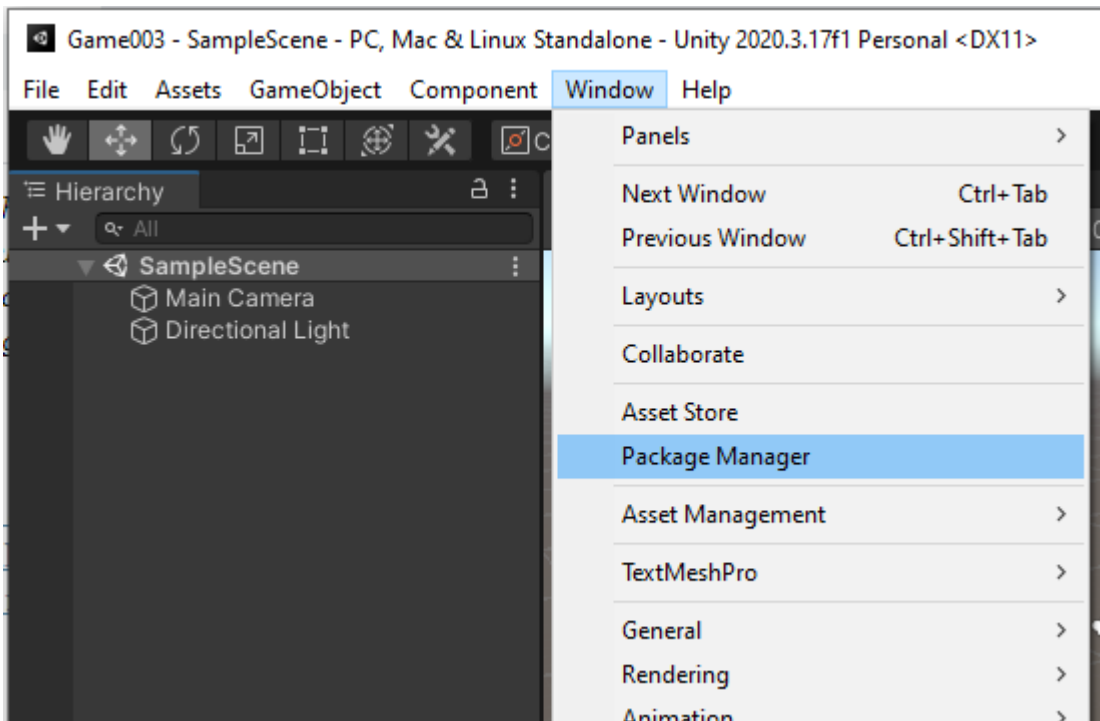
Questa sezione descrive come scaricare, installare e configurare il GameLift plug-in Amazon per Unity, versione 1.0.0.

Prerequisiti

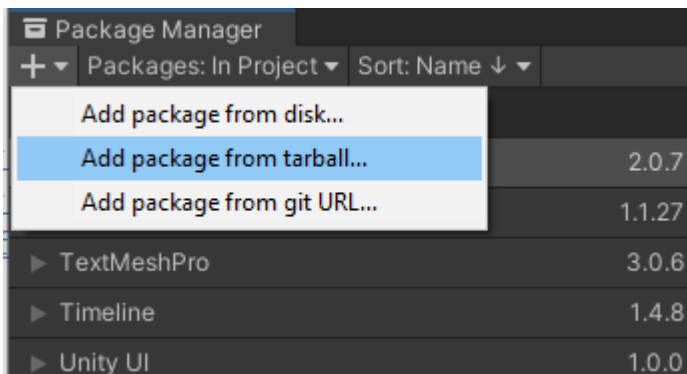
- Unity per Windows 2019.4 LTS, Windows 2020.3 LTS o Unity per macOS
- Versione attuale di Java
- Versione attuale di .NET 4.x

Per scaricare e installare il plugin per Unity

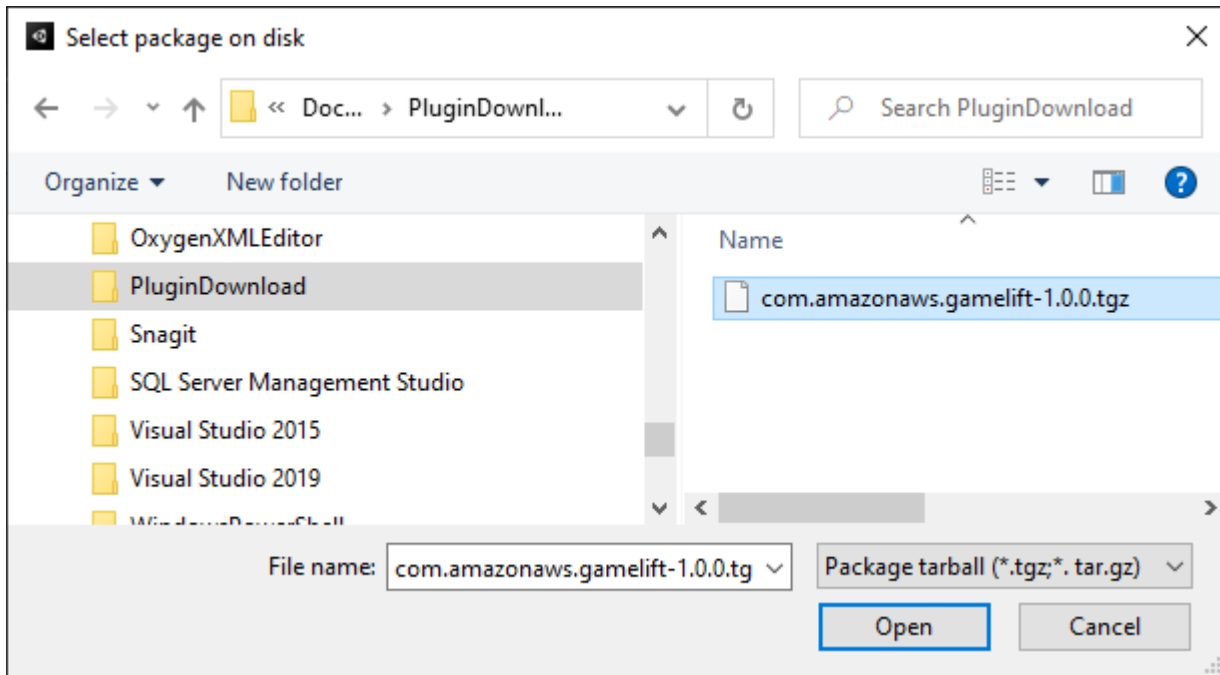
1. Scarica il GameLift plug-in Amazon per Unity. Puoi trovare la versione più recente nella pagina del [repository del GameLift plug-in Amazon per Unity](#). Nella [versione più recente](#), scegli Assets, quindi scarica il `com.amazonaws.gamelift-version.tgz` file.
2. Avvia Unity e scegli un progetto.
3. Nella barra di navigazione in alto, sotto Finestra scegli Package Manager:



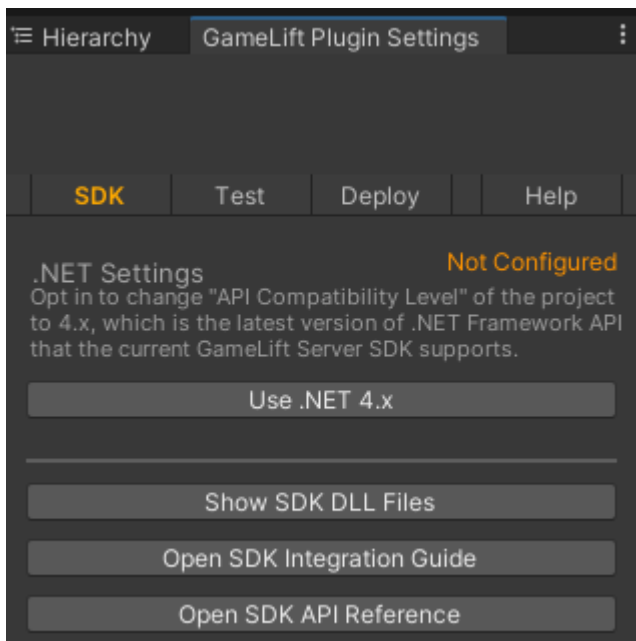
4. Nella scheda Package Manager scegli +, quindi scegli Aggiungi pacchetto da tarball... :



5. Nella finestra Seleziona pacchetti su disco, accedi alla `com.amazonaws.gamelift` cartella, scegli il file `com.amazonaws.gamelift-version.tgz`, quindi scegli Apri:



- Dopo che Unity ha caricato il plug-in, Amazon GameLift appare come una nuova voce nel menu Unity. Potrebbero essere necessari alcuni minuti per installare e ricompilare gli script. La scheda Impostazioni Amazon GameLift Plugin si apre automaticamente.



- 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Usa Amazon GameLift Local per eseguire Amazon GameLift sul tuo dispositivo locale. Puoi utilizzare Amazon GameLift 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 Scarica Amazon GameLift Local. Il plug-in 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 Amazon GameLift Plugin Settings, scegli Amazon GameLift Local Path, vai alla cartella decompressa, scegli il file **GameLiftLocal.jar**, quindi scegli 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 si riferisce al GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza 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 GameLift forniti da Amazon e come utilizzarli.

Prerequisiti

Per implementare lo scenario, è necessario un ruolo IAM per il GameLift servizio Amazon. Per informazioni su come creare un ruolo per Amazon GameLift, consulta [Configura un Account AWS](#).

Ogni scenario richiede le autorizzazioni per le seguenti risorse:

- Amazon GameLift
- Amazon S3
- AWS CloudFormation
- API Gateway
- AWS Lambda
- AWS WAFV2
- Amazon Cognito

Scenari

Note

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

Il GameLift plug-in Amazon 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 GameLift flotta Amazon. 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 le GameLift code di Amazon e un matchmaker personalizzato per raggruppare i giocatori più anziani nel pool di attesa. Crea le seguenti risorse:

- Un argomento di Amazon Simple Notification Service su cui Amazon GameLift pubblica 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 corrispondenze utilizzando le GameLift code Amazon 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 [Guida alla progettazione del GameLift parco veicoli Amazon](#)
- Una GameLift coda Amazon per mantenere alta la disponibilità dei server e ridurre i costi. Per ulteriori informazioni e best practice sulle code, consulta [Progetta una coda per le sessioni di gioco](#)

FlexMatch

Questo scenario utilizza FlexMatch, un servizio di matchmaking gestito, per abbinare i giocatori. Per ulteriori informazioni su FlexMatch, consulta [What is Amazon GameLift FlexMatch](#). 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Il GameLift plug-in Amazon per Unity richiede credenziali di sicurezza per implementare uno scenario. Puoi 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza 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

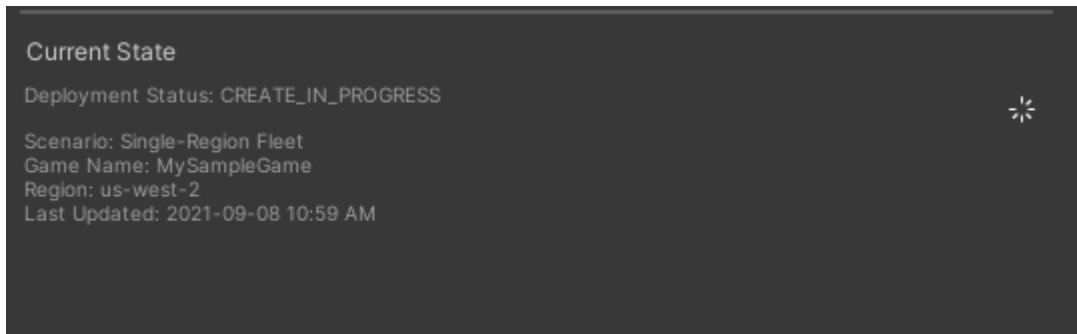
Questo argomento si riferisce al GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Puoi usare uno scenario per testare il tuo gioco con Amazon GameLift. 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 distribuisce lo scenario, Amazon GameLift copia le risorse di gioco nella posizione di bootstrap come parte della distribuzione.

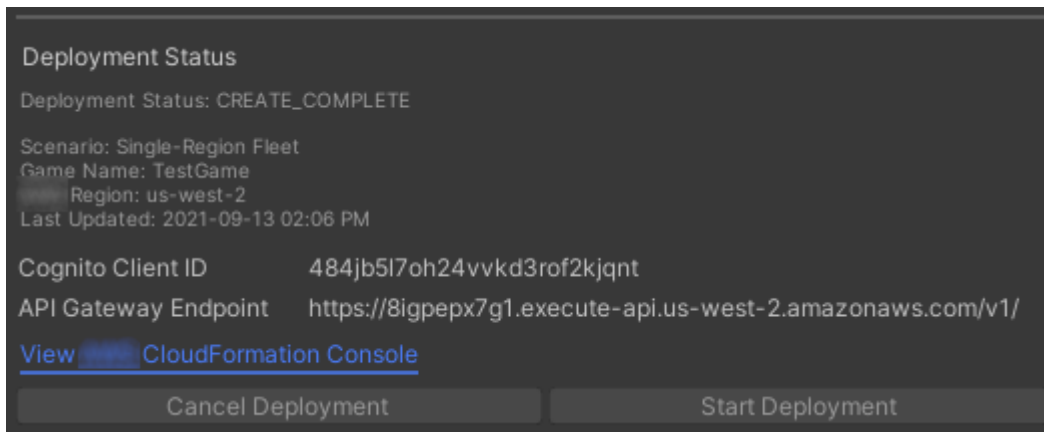
È necessario configurare AWS le credenziali e un AWS account bootstrap per implementare uno scenario.

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 di gioco. Deve essere univoco. Il nome del gioco fa parte del nome AWS CloudFormation dello stack quando distribuisce 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 fino a 30 minuti.



- 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.



- 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.
- Deseleziona la modalità di test locale.
- Inserite l'API Gateway Endpoint e l'ID client Cognito. 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Per eliminare le risorse create per lo scenario, elimina lo stack corrispondente. AWS CloudFormation

Per eliminare le risorse create dallo scenario

1. Nella finestra del GameLift plug-in Amazon 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 cambia in. ROLLBACK_COMPLETE

Integra giochi con Amazon GameLift in Unity

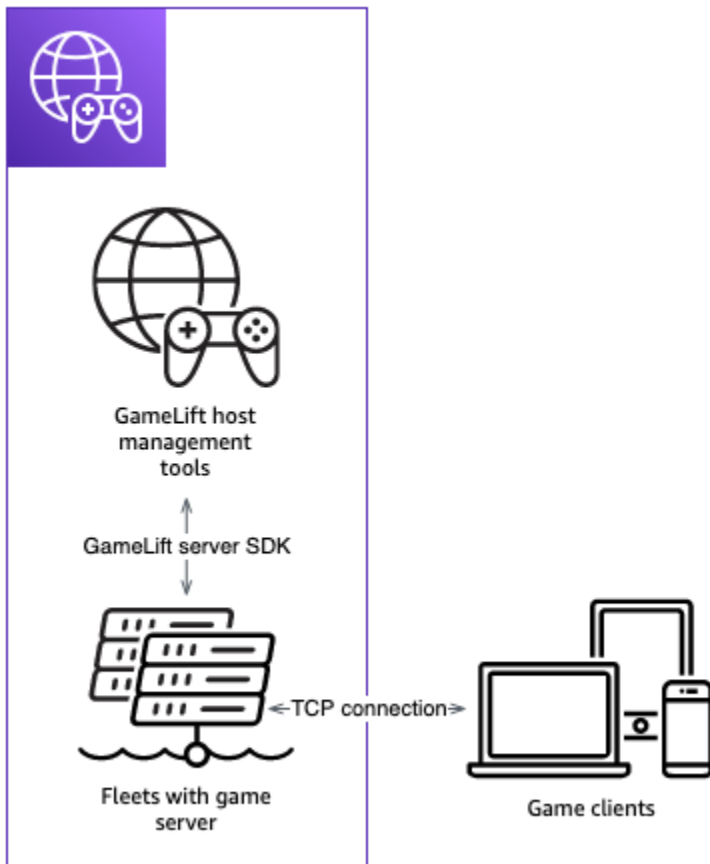
Note

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

Integra il tuo gioco Unity con Amazon GameLift completando le seguenti attività:

- [Integra Amazon GameLift con un progetto di server di gioco Unity](#)
- [Integra Amazon GameLift con un progetto client di gioco Unity](#)

Il diagramma seguente mostra un esempio di flusso di integrazione di un gioco. Nel diagramma, una flotta con il server di gioco viene distribuita su Amazon. GameLift Il client di gioco comunica con il server di gioco, che comunica con Amazon. GameLift



Importa ed esegui un gioco di esempio

i Note

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

Il GameLift plug-in Amazon per Unity include un gioco di esempio che puoi utilizzare per esplorare le basi dell'integrazione del tuo gioco con Amazon. GameLift In questa sezione, crei il client e il server di gioco e poi esegui il test localmente utilizzando Amazon GameLift 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza 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, 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, quindi scegli Applica impostazioni di build del server di esempio di Windows o Applica impostazioni di build del server di esempio di 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 GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza server SDK 4.x o precedente.

Configura i file client di gioco del gioco di esempio.

1. In Unity, nel menu, scegli Amazon GameLift, quindi scegli Applica impostazioni di build del client di esempio di Windows o Applica impostazioni di build del client di esempio di 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 Amazon GameLift Client Settings, 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 scopri come interagisce con Amazon GameLift.

Prova il gioco di esempio a livello locale

Note

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

Esegui il gioco di esempio che hai importato utilizzando Amazon GameLift Local.

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 log di Amazon GameLift Local. La finestra contiene messaggi di registro, inclusi i messaggi inviati tra il server di gioco e Amazon GameLift 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. Nell'Amazon GameLift Sample Game, fornisci un'e-mail e una password, quindi scegli Accedi. L'e-mail e la password non vengono convalidate o utilizzate.
7. In Amazon GameLift Sample Game, scegli Start. 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: playerId
 0 joined
2021-09-15T19:58:0335 PID:20728 Log SERVER: Frame: 8696 Connection accepted: playerId
 1 joined
2021-09-15T19:58:0338 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from playerId 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 playerId 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 Local

```
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/
gss-59f6cc44-4361-42f5-95b5-fdb5825c0f3d
12:56:24,266 INFO || - [WebSocketInvoker] Thread-12 - StartGameSessionRequest:
gameSessionId=arn:aws:gamelift:local::gamesession/fleet-123/
gss-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: gameId: "arn:aws:gamelift:local::gamesession/
fleet-123/gsess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d"
```

Arresta il processo del server

Note

Questo argomento si riferisce al GameLift plug-in Amazon per Unity versione 1.0.0, che utilizza 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.

Integrazione di giochi con il GameLift plug-in Amazon per Unreal Engine

Gli argomenti di questa sezione descrivono il GameLift plug-in Amazon per Unreal Engine (UE) e come utilizzarlo per preparare il tuo progetto di gioco multiplayer per l'hosting con Amazon GameLift. Lavora interamente nel tuo ambiente di sviluppo UE con i flussi di lavoro guidati del plug-in per completare i requisiti di base per l'hosting con Amazon GameLift.

Amazon GameLift è un servizio completamente gestito che consente agli sviluppatori di giochi di gestire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni. Per ulteriori informazioni sull' GameLift hosting Amazon, consulta [Come GameLift funziona Amazon](#).

Argomenti

- [Informazioni sul plugin](#)
- [Workflow del plugin](#)
- [Installa il plugin per Unreal](#)
- [Configura un profilo AWS utente](#)
- [Configura il gioco per testarlo con Amazon GameLift Anywhere](#)
- [Distribuisci il tuo gioco su un hosting cloud con flotte EC2 gestite](#)

Informazioni sul plugin

Il plug-in aggiunge GameLift strumenti e funzionalità di Amazon all'editor UE. I flussi di lavoro guidati del plug-in per GameLift integrare Amazon nel tuo progetto di gioco, designare una workstation come host locale per i test e distribuire il server di gioco sull'hosting cloud Amazon. GameLift

Utilizza le soluzioni di hosting predefinite del plug-in per distribuire il tuo gioco. Configura una flotta Amazon GameLift Anywhere con la tua workstation locale come host. Per l'hosting su cloud, scegli tra due scenari di implementazione comuni che bilanciano la latenza dei giocatori, la disponibilità delle sessioni di gioco e i costi in modi diversi. Uno scenario include un semplice FlexMatch matchmaker e un set di regole. Utilizza queste soluzioni per iniziare rapidamente con una struttura di hosting pronta per la produzione, quindi ottimizza e personalizza secondo necessità.

Il plugin include i seguenti componenti:

- Moduli plug-in per l'editor UE. Una volta installato il plug-in, un nuovo pulsante del menu principale consente di accedere alle GameLift funzionalità di Amazon.
- Librerie C++ per l'API del GameLift servizio Amazon con funzionalità lato client.
- Librerie Unreal per l'SDK GameLift del server Amazon (versione 5).
- 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.

Workflow del plugin

I passaggi seguenti descrivono un approccio tipico all'integrazione e alla distribuzione di un progetto di gioco con il GameLift plug-in Amazon per Unreal Engine. Completi questi passaggi utilizzando l'editor UE e il codice del gioco.

1. Crea un profilo utente che si colleghi al tuo AWS account e fornisca le credenziali di accesso per un utente con account valido con autorizzazioni per utilizzare Amazon. GameLift
2. Aggiungi il codice server al tuo progetto di gioco per stabilire la comunicazione tra un server di gioco in esecuzione e il GameLift servizio con Amazon.
3. Aggiungi il codice client al tuo progetto di gioco che consente ai client di gioco di inviare richieste GameLift ad Amazon per avviare nuove sessioni di gioco e quindi connettersi ad esse.
4. Usa il flusso di lavoro Anywhere per configurare la tua workstation locale come host Anywhere per 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.
5. Usa il flusso di lavoro di hosting EC2 per caricare il tuo server di gioco integrato e implementare una soluzione di hosting su cloud. Quando il server di gioco è pronto, avvia il client di gioco localmente tramite il plug-in, connettiti a una sessione di gioco e gioca.

Quando lavori con il plug-in, creerai e utilizzerai AWS risorse. Queste azioni potrebbero comportare addebiti sull'AWSaccount in uso. Se non l'hai mai fattoAWS, queste azioni potrebbero essere incluse nel piano [AWSgratuito](#).

Installa il plugin per Unreal

Questa sezione descrive le attività di installazione iniziali per aggiungere il plugin a un progetto Unreal Engine. La funzionalità del plug-in è disponibile quando il progetto è aperto nell'editor Unreal.

Note

Puoi utilizzare il GameLift plug-in Amazon con una versione standard dell'editor UE, ma devi utilizzare una versione generata dal codice sorgente quando impacchetti la build del tuo server di gioco.

Prima di iniziare

Ecco cosa ti serve per utilizzare il GameLift plug-in Amazon per Unreal Engine:

- GameLift Plugin Amazon per il pacchetto di rilascio di Unreal Engine. [\[Scarica il sito\]](#).
- Microsoft Visual Studio 2019 o versione successiva.
- Una versione sorgente dell'editor Unreal Engine. È necessaria una versione generata dal codice sorgente per impacchettare i componenti del server per una partita multiplayer. Per maggiori dettagli, inclusi prerequisiti aggiuntivi, consulta la documentazione di Unreal Engine:
 - [Accesso al codice sorgente di Unreal Engine su GitHub](#) You'll need GitHub and Epic Games account.
 - Tutorial sulla [creazione di Unreal Engine dal codice sorgente](#).
- Un progetto di gioco multiplayer con codice di gioco C++. Se lavori con un progetto Blueprint, consulta la documentazione di Unreal su come generare codice sorgente C++ per il tuo progetto.

Aggiungi il plugin al tuo progetto di gioco

Completa le seguenti attività per aggiungere il plugin al tuo progetto di gioco.

Crea l'SDK del server Amazon GameLift C++

1. Decomprimi il pacchetto di rilascio del GameLift plug-in Amazon per Unreal Engine per estrarre due file zip:
 - amazon-gamelift-plugin-unreal-<>-sdk-<>.zip
 - GameLift-Cpp-ServerSDK-<>.zip.

Decomprimi questi file.

2. Apri la GameLift-Cpp-ServerSDK-<> cartella, quindi completa le seguenti istruzioni per la tua piattaforma: Linux o Microsoft Windows.

Linux

1. Esegui i comandi seguenti:

```
mkdir out
cd out
```

```
cmake -DBUILD_FOR_UNREAL=1 ..  
make
```

Questi comandi creano il `/lib/aws-cpp-sdk-gamelift-server.so` file.

2. Copia `/lib/aws-cpp-sdk-gamelift-server.so` nella `amazon-gamelift-plugin-unreal/GameLiftPlugin/Source/GameLiftServer/ThirdParty/GameLiftServerSDK/Linux/x86_64-unknown-linux-gnu/` directory.

Microsoft Windows

1. Esegui i comandi seguenti:

```
mkdir out  
cd out  
cmake -G "Visual Studio 17 2022" -DBUILD_FOR_UNREAL=1 ..  
msbuild ALL_BUILD.vcxproj /p:Configuration=Release
```

Questi comandi generano i seguenti file binari.

- `prefix\bin\aws-cpp-sdk-gamelift-server.dll`
 - `prefix\lib\aws-cpp-sdk-gamelift-server.lib`
2. Copia i file nella `amazon-gamelift-plugin-unreal\GameLiftPlugin\Source\GameLiftServer\ThirdParty\GameLiftServerSDK\Win64\` directory.

Completa le seguenti attività, lavorando sui file di progetto del gioco.

1. Installa i file del plugin.
 - a. Individua la cartella principale del progetto di gioco, ad esempio... `> Unreal Projects/[project-name]/`. Se la cartella `Plugins` non esiste lì, allora creala.
 - b. Vai alla `amazon-gamelift-plugin-unreal` cartella da cui è stato decompresso. `amazon-gamelift-plugin-unreal-<>-sdk-<>.zip` Copia la `GameLiftPlugin` cartella dalla `gamelift-plugin-unreal` cartella alla cartella `Plugins` nella directory del progetto di gioco.
2. Aggiungi il plugin al **.uproject** file.
 - a. Nella cartella principale del progetto di gioco, apri il `.uproject` file.

- b. Aggiorna il file per aggiungere GameLiftPlugin "" e "WebBrowserWidget" alla Plugins sezione e abilitarli. Il codice seguente mostra il .uproject file aggiornato per un gioco chiamato "MyGame».

```
UnrealProjects > MyGame > MyGame.uproject
{
  ...
  "Plugins": [
    {
      "Name": "ModelingToolsEditorMode",
      "Enabled": true,
      "TargetAllowList": [ "Editor" ]
    },
    {
      "Name": "GameLiftPlugin",
      "Enabled": true
    },
    {
      "Name": "WebBrowserWidget",
      "Enabled": true
    }
  ]
}
```

3. Cambia la versione dell'editor UE per il tuo progetto.

Se hai creato un progetto per una versione dell'editor e ora desideri passare a un'altra versione (ad esempio una versione source-build), devi aggiornare il progetto.

Nella cartella principale del progetto di gioco, seleziona il .uproject file e scegli l'opzione Switch Unreal Engine Version. Seleziona una nuova versione dell'editor.

4. Ricostruisci la soluzione del progetto con i tuoi aggiornamenti.
 - a. Nella cartella principale del progetto, cercate un file solution (*.sln). Se non ne esiste nessuno, seleziona il .uproject file e scegli l'opzione Genera file di progetto di Visual Studio.
 - b. Apri il file della soluzione e crea o ricostruisci il progetto.
5. Verifica che il plugin sia abilitato nell'editor UE.

Note

Se hai già l'editor aperto, potresti dover riavviare l'editor prima che riconosca il nuovo plugin.

- a. Apri il progetto nell'editor UE che hai scelto.
- b. Controlla la barra degli strumenti principale dell'editor per il nuovo pulsante del GameLift menu Amazon [immagine necessaria].
- c. Cerca nel Content Browser le risorse del GameLift plug-in Amazon. Assicurati che nelle impostazioni delle opzioni di visualizzazione sia selezionata l'opzione Mostra contenuto del plug-in.

Configura un profilo AWS utente

Dopo aver installato il plug-in, configura un profilo e collegalo a un AWS account utente valido. Puoi mantenere più profili, ma puoi avere solo un profilo attivo alla volta. Ogni volta che lavori nel plugin, seleziona un profilo da usare.

Il mantenimento di più profili ti dà la possibilità di passare da uno scenario di hosting all'altro. Ad esempio, potresti configurare profili con le stesse AWS credenziali ma AWS regioni diverse. Oppure potresti configurare profili con AWS account diversi o con utenti/set di autorizzazioni diversi.

Note

Se hai installato la AWS CLI sulla tua workstation e disponi di un profilo già configurato, il GameLift plug-in Amazon può rilevarlo e lo elencherà come profilo esistente. Il plug-in seleziona automaticamente qualsiasi profilo denominato. [default] È possibile utilizzare un profilo esistente o crearne uno nuovo.

Per gestire i tuoi AWS profili

1. Nella barra degli strumenti principale dell'editor Unreal, scegli il GameLift menu Amazon e seleziona Imposta profili AWS utente. Questa azione apre le impostazioni del progetto per il plugin. Espandi la sezione Profili AWS utente.

2. Se il plugin non rileva un profilo esistente, ti chiede di crearne uno. Puoi creare un nuovo profilo utilizzando un account nuovo o esistente AWS.

Note

È necessario utilizzare la console di AWS gestione per creare un nuovo AWS account e creare o aggiornare un utente con il set di autorizzazioni appropriato.

Quando si configura un profilo, sono necessarie le seguenti informazioni:

- Un account AWS. Se devi creare un nuovo AWS account, segui le istruzioni per creare l'account. Vedi [Creare un AWS account](#) per maggiori dettagli.
 - Un AWS utente con le autorizzazioni per utilizzare Amazon GameLift e altri AWS servizi richiesti. Consulta [Configura un Account AWS](#) le istruzioni sulla configurazione di un utente AWS Identity and Access Management (IAM) con GameLift autorizzazioni Amazon 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. Consulta l'elenco delle [AWS regioni supportate](#).
3. Se il plugin rileva un profilo esistente, non ti viene richiesto di crearne uno. Se desideri aggiornare un profilo o crearne uno nuovo, scegli Gestisci il tuo profilo.

Per avviare il tuo profilo:

Tutti i profili devono essere avviati per essere utilizzati con il plug-in Amazon GameLift . 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.

1. Nella barra degli strumenti principale dell'editor Unreal, scegli l' GameLift icona Amazon e seleziona Imposta profili AWS utente. Questa azione apre le impostazioni del progetto per il plugin. Espandi la sezione Profili AWS utente.

2. Nella sezione Bootstrap il tuo profilo, seleziona un profilo dall'elenco a discesa e controlla lo stato del bootstrap. Se lo stato indica che non esiste alcun bucket, scegli il pulsante Bootstrap e crea profilo per creare un bucket Amazon S3 per il profilo selezionato.
3. Attendi che lo stato del bootstrap passi a «Attivo». Ciò può richiedere alcuni minuti.

Configura il gioco per testarlo con Amazon GameLift Anywhere

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

Per avviare il flusso di lavoro Amazon GameLift Anywhere:

- Nella barra degli strumenti principale dell'editor Unreal, scegli il GameLift menu Amazon 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'AWSaccount 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 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, apporti una serie di aggiornamenti al codice del client e del server per aggiungere funzionalità di hosting. Se non hai ancora configurato una versione sorgente dell'editor Unreal, il plugin fornisce collegamenti alle istruzioni e al codice sorgente.

Con il plugin, puoi sfruttare alcune comodità durante l'integrazione del codice di gioco. Puoi fare un'integrazione minima per configurare le funzionalità di hosting di base. Puoi anche fare un'integrazione personalizzata più ampia. Le informazioni contenute in questa sezione descrivono l'opzione di integrazione minima. Usa le mappe di test incluse nel plug-in per aggiungere la GameLift funzionalità client Amazon al tuo progetto di gioco. Per l'integrazione con il server, usa l'esempio di codice fornito per aggiornare la modalità di gioco del progetto.

Integra la modalità di gioco del tuo server

Aggiungi un codice server al gioco che abiliti la comunicazione tra il server di gioco e il GameLift servizio Amazon. Il tuo server di gioco deve essere in grado di rispondere alle richieste di Amazon GameLift, ad esempio di avviare una nuova sessione di gioco, e anche di segnalare lo stato dello stato del server di gioco e delle connessioni dei giocatori.

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 allinearli al seguente codice di esempio. 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.

```
// 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 "GameLiftServerGameMode.generated.h"

struct FProcessParameters;

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

UCLASS(minimalapi)
class AGameLiftServerGameMode : public AGameModeBase
{
    GENERATED_BODY()
}
```

```
public:
    AGameLiftServerGameMode();

protected:
    virtual void BeginPlay() override;

private:
    void InitGameLift();

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

4. Apri il [project-name]GameMode.cpp file sorgente correlato (ad esempioGameLiftUnrealAppGameMode.cpp). Modificate il codice per allinearlo al codice di esempio seguente. Assicuratevi 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 dei server con Amazon GameLift:

- Inizializza un client Amazon GameLift API. La `InitSDK()` chiamata con i parametri del server è necessaria per una flotta Amazon GameLift 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 GameLift servizio Amazon, tra cui `OnStartGameSessionOnProcessTerminate`, `onHealthCheck`.
- Chiama `ProcessReady()` con una porta designata per avvisare il GameLift servizio Amazon quando sei pronto per ospitare sessioni di gioco.

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

#include "GameLiftServerGameMode.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);

AGameLiftServerGameMode::AGameLiftServerGameMode() :
    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 AGameLiftServerGameMode..."));
}

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

#if WITH_GAMELIFT
    InitGameLift();
#endif
}

void AGameLiftServerGameMode::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.
#endif
}
```

```
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.
        ServerParametersForAnywhere.m_processId =
TCHAR_TO_UTF8(
```

```

        *FText::Format(
            FText::FromString("ProcessId_{0}"),
            FText::AsNumber(std::time(nullptr))
        ).ToString()
    );
}

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);
}

UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_YELLOW);
UE_LOG(GameServerLog, Log, TEXT(">>>> Web Socket 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, 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, GameLift 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. GameLift 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 GameLift 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.
//GameLift 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. GameLift
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 = FString::Atoi(*Value);
        }
    }
}

//Here, the game server tells GameLift where to find game session log files.
//At the end of a game session, GameLift 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 GameLift 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 plug-in 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 flotta EC2 gestita. Quando gestisci il tuo client, puoi scegliere di connetterti a entrambe le flotte.

- Le funzionalità del client includono la ricerca di una sessione di gioco `SearchGameSessions (())`, la creazione di una nuova sessione di gioco `CreateGameSession (())` e la partecipazione diretta 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 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.

Costruisci i componenti del tuo gioco

1. Crea nuovi file di destinazione per server e client
 - a. Nella cartella del progetto di gioco, vai alla cartella `Source` e trova i `Target.cs` file.
 - b. Copia il file `[project-name]Editor.Target.cs` in due nuovi file denominati `[project-name]Client.Target.cs` e `[project-name]Server.Target.cs`.
 - c. Modifica ciascuno dei nuovi file per aggiornare i valori del nome della classe e del tipo di destinazione, come mostrato:

```
UnrealProjects > MyGame > Source > MyGameClient.Target.cs
// Copyright Epic Games, Inc. All Rights Reserved.
```

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

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

```
UnrealProjects > MyGame > Source > MyGameServer.Target.cs
// Copyright Epic Games, Inc. All Rights Reserved.

using UnrealBuildTool;
using System.Collections.Generic;

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

2. Aggiorna il .Build.cs file.

- a. Apri il .Build.cs file del tuo progetto. Questo file si trova in UnrealProjects/[project name]/Source/[project name]/[project name].Build.cs.
- b. Aggiornate la ModuleRules classe come illustrato nel seguente esempio di codice.

```
public class MyGame : ModuleRules
{
    public GameLiftUnrealApp(TargetInfo Target)
    {
```

```
PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject",
"Engine", "InputCore" });
bEnableExceptions = true;

if (Target.Type == TargetRules.TargetType.Server)
{
    PublicDependencyModuleNames.AddRange(new string[]
{ "GameLiftServerSDK" });
    PublicDefinitions.Add("WITH_GAMELIFT=1");
}
else
{
    PublicDefinitions.Add("WITH_GAMELIFT=0");
}
}
}
```

3. Ricostruisci la soluzione del tuo progetto di gioco.
4. Apri il tuo progetto di gioco in una versione originale dell'editor di Unreal Engine.
5. Fai quanto segue sia per il client che per il 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.

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'AWSaccount 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 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.

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. 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 verifica di potersi connettere al computer e restituisce informazioni sull'elaborazione appena registrata. Crea anche gli argomenti della console necessari agli eseguibili di gioco per inizializzare la comunicazione con il servizio Amazon. GameLift

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 GameLift servizio. Il plug-in genera e memorizza automaticamente un token di autenticazione per la flotta Anywhere ogni volta che avvia 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.

Non è necessario eseguire alcuna azione in questo passaggio.

Passaggio 6: Avvia il gioco

A questo punto, hai completato tutte le attività necessarie per avviare e giocare al gioco multiplayer su una workstation locale utilizzando Amazon GameLift.

1. Avvia il tuo server di gioco. Il server di gioco avviserà Amazon GameLift quando sarà pronto per ospitare 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 ad Amazon GameLift tramite il nuovo servizio di backend. In risposta GameLift, Amazon chiama il server di gioco, in esecuzione sul tuo computer locale, per avviare una nuova sessione di gioco. Quando la sessione di gioco è pronta per accettare giocatori, Amazon GameLift fornisce le informazioni di connessione per consentire al client di gioco di partecipare alla sessione di gioco.

Distribuisci il tuo gioco su un hosting cloud con flotte EC2 gestite

In questo flusso di lavoro, utilizzi il plug-in per modificare il gioco per l'hosting su risorse di elaborazione basate su cloud gestite da Amazon. GameLift Aggiungi il codice di gioco client e server per le GameLift funzionalità di Amazon, quindi carica la build del server sul GameLift servizio Amazon per la distribuzione nelle risorse basate sul cloud. 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 flusso di lavoro Amazon EC2 GameLift gestito da Amazon:

- Nella barra degli strumenti principale dell'editor Unreal, scegli il GameLift menu Amazon 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.

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, apporti una serie di aggiornamenti al codice del client e del server per aggiungere funzionalità di hosting. Se non hai ancora configurato una versione sorgente dell'editor Unreal, il plugin fornisce collegamenti alle istruzioni e al codice sorgente.

Se hai integrato il gioco per utilizzarlo con una flotta Anywhere, non devi apportare alcuna modifica al codice del gioco. Se utilizzi la mappa di gioco di avvio, questa funziona anche con le implementazioni EC2.

- [Imposta il codice di gioco \(ovunque\)](#)
- [Costruisci i componenti del tuo gioco](#)

Dopo aver creato il tuo server di gioco, completa le seguenti attività per prepararlo per il caricamento su Amazon GameLift.

Per impacchettare il tuo server, crea un pacchetto per la distribuzione su cloud

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

1. Copia lo script di installazione, incluso nel download del plugin, nella radice della `WindowsServer` cartella. Cerca il file `[project-name]/Plugins/Resources/CloudFormation/extra_server_resources/install.bat`. Amazon GameLift utilizza questo file per installare la build del server su ogni risorsa di hosting EC2.
2. Copia il `VC_redist.x64.exe` file, incluso nell'installazione di Visual Studio, nella radice della `WindowsServer` cartella. Questo file si trova in genere in `C:/Program Files (x86)/Microsoft Visual Studio/2019/Professional/VC/Redist/MSVC/v142`.
3. Copia le DLL OpenSSL per il tuo server di gioco integrato nella cartella `WindowsServer/MyGame/Binaries/Win64`. Assicurati che le DLL siano per la stessa versione utilizzata nella build del server. Copia i seguenti file:
 - `libssl-3-x64.dll`
 - `libcrypto-3-x64.dll`

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:
 - AWSfleet (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 alle API.
 - WebACL 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 sedi. Include due flotte Spot e una flotta On-Demand come backup.
 - AWSCode 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 alle API.

- WebACL 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 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, distribuisce il gioco su una soluzione di hosting cloud in base allo scenario di distribuzione scelto. Questo processo può richiedere fino a 40 minuti per 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 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'istanza AWS EC2. Almeno un processo 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 su Amazon GameLift. 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.

Ottieni dati sulla flotta per un'GameLiftistanza Amazon

In alcune situazioni, la creazione di un gioco personalizzato o lo script Realtime Servers possono richiedere informazioni sulla GameLift flotta Amazon. Ad esempio, la build o lo script del gioco potrebbero includere codice per:

- Monitora l'attività in base ai dati della flotta.
- Raggruppa le metriche per monitorare 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 il matchmaking, la scalabilità aggiuntiva della capacità o i 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 GameLift della flotta Amazon](#).

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",
  "fleetName": "ReallyFunGameTestFleet08",
  "fleetType": "ON_DEMAND",
  "instanceRoleArn": "arn:aws:iam::123456789012:role/S3AccessForGameLift",
  "instanceType": "c5.large",
  "serverLaunchPath": "/local/game/reallyfungame.exe"
}
```

Aggiungere il FlexMatch matchmaking

Usa Amazon GameLift FlexMatch per aggiungere la funzionalità di matchmaking dei giocatori ai tuoi giochi GameLift ospitati su Amazon. Puoi utilizzarlo FlexMatch con server di gioco personalizzati o server in tempo reale.

FlexMatch associa il servizio di abbinamento a un motore di regole personalizzabili. Decidi come abbinare i giocatori in base agli attributi dei giocatori e alle modalità di gioco più adatte al tuo gioco. FlexMatch gestisce i dettagli della valutazione dei giocatori in cerca di una partita, della formazione di partite con una o più squadre e dell'avvio delle 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 utilizza le code per individuare le migliori sedi di hosting possibili per i giochi in più regioni e tipi di computer. In particolare, GameLift le code di Amazon possono utilizzare i dati di latenza, se forniti dai client di gioco, per inserire sessioni di gioco in modo che i giocatori sperimentino 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 [Amazon GameLift FlexMatch Developer Guide](#):

- [Come GameLift FlexMatch funziona Amazon](#)
- [FlexMatch fasi di integrazione](#)

Gestione dell'hosting con GameLift contenitori Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Amazon GameLift fornisce un servizio di hosting cloud completo per supportare soluzioni containerizzate per l'hosting di server di gioco. Con le flotte di GameLift container di Amazon, puoi sfruttare i vantaggi dei container come la portabilità, l'agilità e la tolleranza ai guasti.

Funzionalità principali

Le seguenti funzionalità sono disponibili con le flotte di GameLift container Amazon.

- Sviluppa un'architettura di container personalizzata con contenitori leggeri per eseguire il software del tuo server di gioco sulle risorse di GameLift hosting di Amazon.
- Includi Amazon GameLift Agent per gestire il ciclo di vita dei processi dei server di gioco all'interno dei tuoi container. L'agente on-compute esegue le tue istruzioni su quando e come avviare i processi del server e su quanti devono essere mantenuti per l'hosting delle sessioni di gioco.
- Personalizza le risorse fornite da Amazon GameLift per creare immagini di container con la tua applicazione per server di gioco. Usa il dockerfile fornito per creare un'immagine del contenitore basata su Linux. Archivia le immagini delle tue flotte di container in un repository privato Amazon Elastic Container Registry (Amazon ECR).
- Offri ai giocatori esperienze a bassa latenza distribuendo le risorse della flotta di container in qualsiasi zona locale Regione AWS supportata da Amazon. GameLift Crea flotte di container con più sedi per una gestione semplificata della flotta. Per informazioni, consulta [Sedi GameLift di hosting Amazon](#).
- Metti alla prova le tue soluzioni di hosting di giochi containerizzate con una flotta Amazon GameLift Anywhere. Usa Anywhere per testare localmente lo sviluppo della tua soluzione, inclusa l'integrazione con Amazon GameLift SDK e le configurazioni dell'immagine del contenitore.
- 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.
- Usa gli strumenti di posizionamento delle sessioni di GameLift gioco di Amazon, tra cui code e FlexMatch matchmaking, per abbinare i giocatori alle migliori sessioni di gioco possibili ospitate sulle tue flotte di container.

- Gestisci le risorse della flotta di container utilizzando AWS CloudFormation modelli per Amazon GameLift.

Utilizzo di flotte di container durante l'anteprima pubblica

La nuova funzionalità relativa alle flotte di container è attualmente disponibile in anteprima pubblica. Durante questa fase, sono supportate le seguenti GameLift funzionalità di Amazon:

- Usa flotte di container per ospitare server di gioco creati per Linux. Le flotte di container utilizzano `Amazon_Linux_2023` e supportano immagini di container Linux. I contenitori Windows non sono supportati.
- Integra progetti di server di gioco solo con Amazon GameLift Server SDK versione 5+. Le versioni precedenti non sono supportate.
- Utilizza uno qualsiasi dei tipi di istanza Amazon EC2 On-Demand supportati da Amazon. GameLift Le flotte Spot non sono supportate in questo momento.

Come funzionano i container in Amazon GameLift

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Le flotte di GameLift container Amazon sono progettate per offrire flessibilità nel modo in cui distribuisce 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 Questo argomento descrive gli elementi strutturali di base per far funzionare i container su una flotta GameLift gestita da Amazon, illustra le architetture comuni e delinea alcuni concetti fondamentali.

Componenti della flotta di container

Parco istanze

Una flotta di container è una raccolta di istanze Amazon EC2, gestite da Amazon GameLift, che eseguono i tuoi server di gioco containerizzati. Quando crei una flotta, configuri il modo in cui l'architettura dei container e il software del server di gioco vengono distribuiti su ciascuna istanza della flotta. Puoi distribuire una flotta di container in una Regione AWS o più località geografiche.

Puoi utilizzare gli strumenti di ridimensionamento GameLift manuale o automatico di Amazon per scalare la capacità di una flotta di container di ospitare sessioni di gioco e giocatori.

Istanza

Un'istanza Amazon EC2 è il server virtuale che fornisce capacità di elaborazione per l'hosting dei giochi. Con Amazon GameLift, puoi scegliere tra diversi 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 distribuisce le istanze 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 utilizza il concetto di gruppo di contenitori per descrivere e gestire un set di contenitori. Un gruppo di contenitori è simile a un contenitore «task» o «pod». All'interno di ciascun gruppo di contenitori, è possibile definire in che modo i contenitori condividono le risorse di CPU e memoria disponibili. È inoltre possibile impostare dipendenze tra i contenitori e gestire il ciclo di vita del gruppo di contenitori.

I gruppi di container possono essere replicati su ogni istanza del parco veicoli per ottimizzare l'uso delle risorse. È possibile gestire la replica impostando la strategia di pianificazione di un gruppo di contenitori, come segue:

- I gruppi di contenitori di replica gestiscono i contenitori che eseguono l'applicazione del server di gioco e il software di supporto. Tutte le flotte di container devono definire un gruppo di contenitori di replica. Un gruppo di repliche può avere più copie su ogni istanza del parco veicoli, a seconda dei requisiti del gruppo di contenitori e delle risorse del tipo di istanza in uso. Tutti i contenitori del gruppo di repliche si scalano automaticamente insieme su un'istanza.
- I gruppi di contenitori Daemon, che sono opzionali, potrebbero essere utili per l'esecuzione di 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 demoni. I gruppi di container daemon non vengono replicati: ogni istanza della flotta ha al massimo una copia del gruppo di daemon. Ciò significa che i container di un gruppo di daemon non sono scalabili su un'istanza della flotta insieme ai contenitori di un gruppo di repliche.

Una flotta di container deve avere un gruppo di contenitori di repliche e può facoltativamente avere un gruppo di daemon.

Container

Il contenitore è l'elemento più basilare di un'architettura basata su contenitori. È costituito da un'immagine del contenitore con eseguibili software e file dipendenti. Quando definisci un contenitore da utilizzare con Amazon GameLift, configuri il modo in cui il software viene eseguito nel contenitore.

Ogni gruppo di container in una flotta di container deve avere 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.

I tipi di contenitore includono:

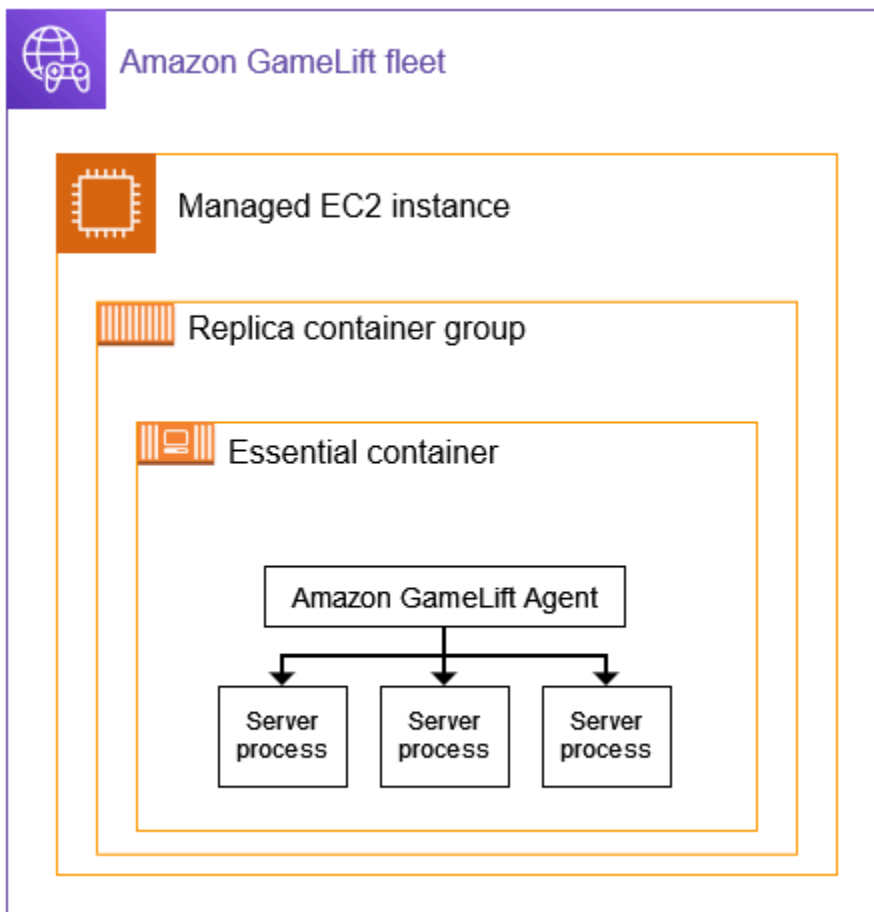
- Essential Replica Container include tutto il necessario per eseguire i processi del server di gioco e ospitare sessioni di gioco per i giocatori. Include la build del server di gioco, che è integrata con l'SDK del GameLift server Amazon, e il software dipendente. Include anche Amazon GameLift Agent, che gestisce il ciclo di vita dei processi del server di gioco. Il gruppo di container di replica di una flotta ha esattamente un container di replica essenziale.
- I contenitori di replica non essenziali, detti anche contenitori «sidecar», eseguono software per supportare l'applicazione del server di gioco. L'utilizzo di un contenitore sidecar ti consente di eseguire e scalare il software di supporto insieme ai server di gioco, ma gestirlo come contenitori separati. Se questo tipo di contenitore si guasta, viene riavviato solo il contenitore stesso; il gruppo di contenitori non ne risente.
- I contenitori Daemon eseguono un servizio daemon per gestire i processi in background. Un uso comune di un contenitore daemon consiste nell'eseguire un [agente Amazon CloudWatch \(CloudWatch\)](#) per raccogliere metriche, log e tracce per i contenitori. I contenitori daemon possono essere essenziali o non essenziali a seconda del momento in cui un errore del contenitore deve comportare il riavvio del gruppo di contenitori.

Calcolo

Un computer è una risorsa di hosting della flotta registrata con il GameLift servizio Amazon ed è in grado di comunicare con il servizio. In una flotta di container, un calcolo è un contenitore con un processo che gestisce il processo di registrazione dell'elaborazione. Nel contenitore di replica essenziale di una flotta di container, Amazon GameLift Agent registra automaticamente questo contenitore come elaborazione.

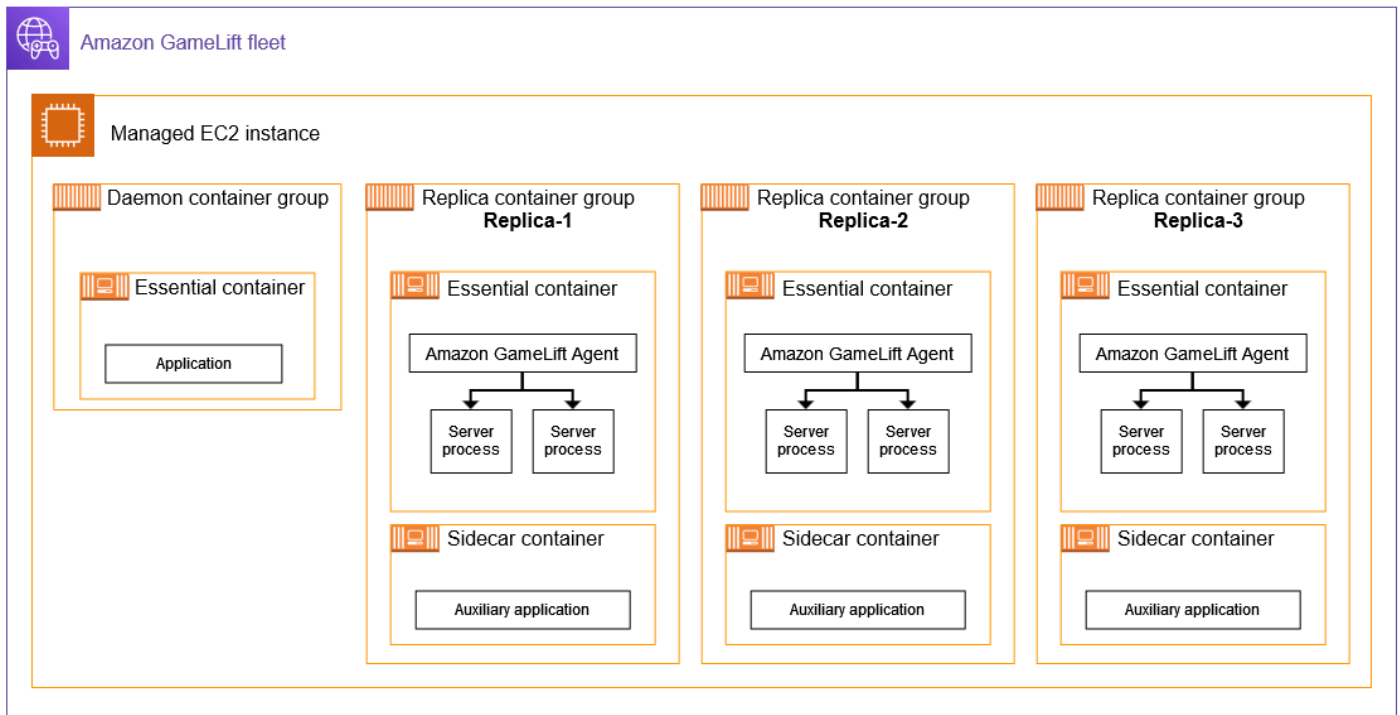
Architetture comuni

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 container di replica. Il gruppo di contenitori dispone di un unico contenitore essenziale che esegue Amazon GameLift Agent, l'applicazione server di gioco e tutto il software di supporto per ospitare sessioni di gioco. L'agente implementa istruzioni specifiche del parco macchine per eseguire tre processi server contemporaneamente. Poiché esiste un gruppo di contenitori di repliche per istanza, ogni istanza del parco istanze esegue tre processi server contemporaneamente.



Questo secondo esempio illustra una progettazione più complessa di una flotta di container. In questo esempio, la flotta ha un gruppo di container di replica con più container e un gruppo di container daemon con un solo container. La configurazione della flotta inserisce tre copie del gruppo di contenitori di replica in ogni istanza della flotta. Il gruppo di contenitori daemon non viene mai replicato.

Ogni set di contenitori del gruppo di repliche del gruppo ha tre copie su ogni istanza. In ogni contenitore di replica essenziale, all'agente viene richiesto di eseguire due processi server contemporaneamente. Di conseguenza, ogni istanza del parco istanze esegue sei processi server contemporaneamente (due processi in ciascuno dei tre contenitori di replica essenziali).



Concetti principali

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

Imballaggio per gruppi di container

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, trova il maggior numero di gruppi di container di repliche che puoi collocare su un'istanza della flotta senza influire sulle prestazioni del server di gioco.

Amazon GameLift può aiutarti a farlo. Calcola il numero massimo di gruppi di repliche per istanza, in base alle seguenti informazioni:

- Il tipo di istanze del parco istanze e le risorse di CPU e memoria disponibili.

- I requisiti di CPU e memoria impostati per tutti i contenitori del gruppo di repliche.

I requisiti di CPU e memoria impostati per tutti i contenitori di un gruppo di daemon, se presente.

- Risorse riservate per gestire contenitori e altre applicazioni critiche su ogni istanza.

Quando crei una flotta di container, puoi scegliere di utilizzare il numero massimo calcolato oppure puoi sovrascrivere il numero calcolato specificando il numero desiderato. Come best practice, sperimenta il software del tuo server di gioco containerizzato per determinare con precisione i requisiti di risorse. Usa questi dati per trovare una strategia di imballaggio ottimale per le prestazioni dei server di gioco.

Server di gioco e Amazon GameLift Agent

Quando crei il tuo contenitore di replica essenziale, impacchetti il software del server di gioco e Amazon GameLift Agent insieme nella stessa immagine del contenitore. Questo agente on-computing controlla il ciclo di vita dei server di gioco nel container. In ogni gruppo di contenitori di repliche, il contenitore di replica essenziale esegue l'agente e tutti i processi del server di gioco.

Amazon GameLift Agent esegue le istruzioni nella configurazione di runtime della flotta di container. La configurazione di runtime identifica (1) l'eseguibile da avviare, (2) un set opzionale di parametri di avvio e (3) il numero di processi da eseguire contemporaneamente. Una configurazione di runtime può contenere istruzioni per più eseguibili diversi. Almeno un'istruzione deve riguardare l'eseguibile del server di gioco. Ad esempio, una configurazione di runtime potrebbe indicare all'agente di mantenere 10 processi del server di gioco eseguibili per l'uso in produzione, 1 processo dello stesso eseguibile con parametri di avvio speciali per i test e 1 processo per un'utilità di registrazione.

Puoi modificare la configurazione di runtime di una flotta in qualsiasi momento. L' GameLift agente Amazon richiede periodicamente aggiornamenti dal servizio. Quando è disponibile una configurazione di runtime aggiornata, l'agente la riceve e inizia a implementare le istruzioni. Le azioni possono includere l'aggiunta o la chiusura dei processi del server.

Amazon GameLift Agent è una versione open source dell'agente on-computing che Amazon GameLift utilizza per le flotte EC2 gestite. Questa guida fornisce istruzioni su come creare l'agente dal codice sorgente e trasformarlo in un'immagine del contenitore. L'agente gestisce le seguenti attività:

Gestione dei processi del server:

- Avvia, spegni e sostituisci i processi del server in base alla configurazione di runtime.

- Arresta i processi del server quando non si attivano in tempo.
- Segnala ad Amazon GameLift quando un processo server termina.
- Emetti eventi di flotta per i processi del server.

Gestione dei container:

- Arresta i processi del server in risposta alle richieste di Amazon. GameLift
- Segnala lo stato del contenitore.

Attività di caricamento dei registri:

- Carica i log delle sessioni di gioco in un bucket Amazon S3 designato.
- Carica i log degli agenti in elaborazione in un bucket Amazon S3 designato.

Dimensionamento della capacità del parco istanze

La capacità della flotta misura il numero di sessioni di gioco che la flotta può ospitare contemporaneamente. Puoi anche misurare la capacità in base al numero di giocatori che la flotta può supportare contemporaneamente.

Per aumentare o diminuire la capacità di hosting di una flotta, aggiungi o rimuovi le istanze della flotta. La strategia di imballaggio di una flotta di container determina quante sessioni di gioco vengono eseguite contemporaneamente su ciascuna istanza della flotta. Questo numero indica il numero di sessioni di gioco (e di slot per giocatori) che aggiungi o sottrai quando aumenti o diminuisce la capacità della flotta.

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

- Imposta manualmente la capacità della flotta impostando un numero specifico di istanze della flotta desiderato.
- Imposta il ridimensionamento automatico selezionando come target il buffer desiderato di istanze disponibili (tracciamento del target). Questo metodo mantiene automaticamente una serie di risorse di hosting inattive in modo che i giocatori in arrivo possano sempre iniziare a giocare rapidamente. 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).

Connessioni client/server di gioco

Le flotte EC2 gestite e le flotte di container gestiscono le connessioni tra client di gioco e server di gioco ospitati sul cloud in modo simile. Quando Amazon GameLift crea una nuova sessione di gioco, il servizio comunica le informazioni di connessione della sessione di gioco. I client di gioco utilizzano le informazioni per connettersi direttamente al server di gioco che ospita la sessione di gioco. Per tutti i tipi di flotte, le informazioni di connessione consistono nell'assegnazione di un indirizzo IP e di una porta.

Quando si crea una flotta di container, si definiscono due set di intervalli di porte. Innanzitutto, definisci una serie di porte di connessione rivolte verso l'esterno che consentono ai client di gioco di connettersi a un gioco. In secondo luogo, definisci un set di porte container solo interne, che vengono assegnate a ciascun processo del server di gioco in esecuzione nel contenitore. Amazon mappa GameLift dinamicamente le porte interne dei container alle porte di connessione esterne per consentire ai giocatori di accedere ai giochi. Questo approccio fornisce un ulteriore livello di sicurezza proteggendo i server di gioco dall'accesso diretto alle porte dei container.

Quando si definiscono gli intervalli di porte per una flotta di container, è necessario fornire intervalli con un numero di porte sufficiente per ospitare tutti i processi del server eseguiti contemporaneamente sui container di un'istanza.

Per un ulteriore controllo, puoi anche impostare le autorizzazioni in entrata per una flotta. Le autorizzazioni in entrata determinano quali porte di connessione sono aperte per il traffico in entrata. Puoi modificare le autorizzazioni in entrata di un parco veicoli in qualsiasi momento. Con le autorizzazioni in entrata puoi chiudere rapidamente tutte le porte di connessione, aprirne alcune o aprirle tutte secondo necessità.

Roadmap di sviluppo per contenitori Amazon GameLift

Questa documentazione riguarda una funzionalità che è in versione di anteprima pubblica. ed è soggetta a modifiche.

Il seguente flusso di lavoro riassume i passaggi per far funzionare i server di gioco su una flotta di GameLift container Amazon.

Passaggio 1: integra il tuo gioco con Amazon GameLift

Aggiungi funzionalità al tuo server di gioco in modo che possa comunicare con il GameLift servizio Amazon quando viene distribuito su una flotta di container. Se utilizzi il FlexMatch matchmaking, aggiungi questa funzionalità al server e al client di gioco. Per informazioni dettagliate, consulta [Integra il tuo gioco con Amazon GameLift](#).

- Ottieni l'SDK GameLift del server Amazon (versione 5+) e configuralo con il tuo progetto di gioco. 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 richiesta.
- Package della build del tuo server di gioco per Linux. Se stai sviluppando su Windows, questo passaggio potrebbe richiedere del lavoro aggiuntivo per configurare un ambiente Linux.
- (Facoltativo) Testa l'integrazione del tuo server di gioco utilizzando una GameLift Anywhere flotta Amazon. Esegui il test prima di preparare l'immagine del contenitore per isolare i problemi relativi al tuo lavoro di integrazione. Per testare le connessioni client/server di gioco, integra anche il tuo client di gioco.

Note

Se stai sviluppando su Windows, configura uno spazio di lavoro Linux separato o usa 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.

Passaggio 2: prepara l'immagine del contenitore del server di gioco

Crea un'immagine del contenitore che esegua i processi del server di gioco e archivala in un repository Amazon Elastic Container Registry (Amazon ECR) da utilizzare con Amazon. GameLift Per istruzioni dettagliate, vedi [Prepara un'immagine del contenitore con il software del tuo server di gioco](#).

- Configura una directory di lavoro per l'immagine del contenitore, con la build del gioco Linux, lo script di installazione e tutto il software e le dipendenze di supporto.
- Ottieni il codice sorgente di Amazon GameLift Agent, compilalo e aggiungi il `jar` file alla tua directory di lavoro.
- Ottieni il Dockerfile predefinito e modificalo per configurare un'immagine del contenitore con il software del tuo server di gioco.
- Crea l'immagine del tuo contenitore. Esegui questo passaggio in un ambiente Linux.

- Crea un repository privato Amazon ECR e inviati l'immagine del contenitore. Crea il repository nello stesso Account AWS luogo in Regione AWS cui intendi distribuire la tua flotta di container.
- (opzionale) Testa le immagini dei container utilizzando la tua Anywhere flotta. Puoi impostare una configurazione di runtime per passare istruzioni ad Amazon GameLift Agent.

Fase 3: Crea i tuoi contenitori e i tuoi gruppi di contenitori

Progetta un'architettura di container per l'hosting di giochi su Amazon GameLift. Consulta [Progetta una flotta di GameLift container Amazon](#) e [Crea definizioni di gruppi di container per una flotta di GameLift container Amazon](#).

- Definisci le configurazioni dei tuoi container. Per ogni contenitore, definirai problemi come i processi di runtime, l'allocazione della memoria, i controlli di integrità, le porte di rete, ecc.
- Utilizza la GameLift console Amazon o la AWS CLI per creare definizioni di gruppi di contenitori con le configurazioni dei contenitori. Quando crei una definizione di gruppo di contenitori, Amazon GameLift scatta un'istantanea di ogni immagine del contenitore in quel momento.

Passaggio 4: distribuisci il tuo server di gioco containerizzato su una flotta di container

Utilizza le definizioni dei gruppi di container create nel passaggio precedente per creare una flotta di container e distribuire il software del server di gioco containerizzato. Per informazioni, consulta [Crea una flotta di GameLift container Amazon](#).

- Usa la GameLift console Amazon o la AWS CLI per creare una flotta di container.
- Tieni traccia dello stato della flotta man mano che le istanze della flotta vengono distribuite e attivate. Controlla gli eventi di creazione della flotta per verificare che la flotta venga distribuita correttamente in tutte le sedi.
- Verifica che i client di gioco possano richiedere e partecipare alle sessioni di gioco e giocare. Se hai impostato il matchmaking, prova questi scenari.

Fase 5: Gestisci le tue flotte

Mentre ti prepari per l'utilizzo a livello di produzione, crea la tua soluzione di hosting di giochi e gestisci il ciclo di vita dell'hosting.

- Crea flotte con più sedi e altre flotte per supportare la tua base di giocatori. Regioni AWS
- Configura il posizionamento degli host di giochi con code o matchmaking. FlexMatch Consulta queste risorse:
 - [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#)
 - [FlexMatch Developer Guide](#)

- Imposta la scalabilità automatica per gestire la capacità della flotta in base alla domanda dei giocatori per le sessioni di gioco.
- Imposta il monitoraggio per le tue flotte di container. Lavora con i GameLift parametri di Amazon, recupera i log delle sessioni di gioco e i log dei container, configura l'accesso remoto ai singoli contenitori.
- Imposta la gestione a lungo termine delle flotte di container. Utilizza gli alias delle flotte per semplificare il processo di aggiornamento delle flotte di container. Crea AWS CloudFormation modelli per gestire il ciclo di vita della flotta. Consulta queste risorse:
 - [Aggiungi un alias a una flotta Amazon GameLift](#)
 - [Gestisci le risorse utilizzando AWS CloudFormation](#)

Integra il tuo gioco con Amazon GameLift

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Prima di poter creare un'immagine contenitore con il software del server di gioco e distribuirla su Amazon GameLift per l'hosting su cloud, integra il progetto di gioco con l'SDK del GameLift server Amazon e crea un server di gioco da eseguire su Linux. Questo argomento presenta i vari strumenti di integrazione GameLift forniti da Amazon.

I server di gioco ospitati devono essere in grado di comunicare con il GameLift servizio Amazon. Imposta la comunicazione aggiungendo l'SDK del GameLift server Amazon (versione 5+) al tuo progetto di gioco e modificando il codice del server del gioco. Amazon GameLift fornisce risorse e documentazione SDK per server per supportare diverse lingue e motori di gioco.

Il processo di integrazione per i server di gioco containerizzati è praticamente identico all'integrazione dei server di gioco per l'hosting su flotte gestite EC2 o Amazon. GameLift Anywhere

Strumenti di integrazione

Amazon GameLift fornisce i seguenti strumenti e supporto linguistico per l'integrazione:

Per gli sviluppatori di Unreal Engine

Usa il plugin leggero per Unreal. Questo plugin include le librerie SDK del server C++ con le funzionalità Amazon GameLift richieste. Usa la documentazione per configurare il tuo progetto di

gioco Unreal per il plug-in e aggiorna il codice di gioco con i blocchi di codice forniti per aggiungere le funzionalità richieste per le build di server e client.

- [Scarica il plugin SDK](#)
- [Guida: integra il tuo progetto Unreal con Amazon GameLift](#)
- [Guida di riferimento: C++ Server SDK 5 per Unreal](#)

Nota: il plug-in GameLift standalone Amazon per Unreal Engine non supporta l'uso di flotte di container.

Per gli sviluppatori di Unity

Usa il plugin leggero per Unity. Questo plugin include le librerie SDK del server C# con le funzionalità Amazon GameLift richieste. Usa la documentazione per configurare il tuo progetto di gioco Unreal per il plug-in e aggiorna il codice di gioco con i blocchi di codice forniti per aggiungere le funzionalità richieste per le build di server e client.

- [Scarica il plugin SDK](#)
- [Guida: integra il tuo progetto Unity con Amazon GameLift](#)
- [Guida di riferimento: C# Server SDK 5 per Unity](#)

Nota: il plug-in GameLift standalone Amazon per Unity non supporta l'uso di flotte di container.

Per sviluppatori che utilizzano altri motori di gioco

Segui questa guida generale all'integrazione di server e client:

- [Integra un server di gioco](#)
- [Integra un client di gioco](#)

Amazon GameLift offre librerie SDK 5 per server per le seguenti lingue:

- [Server SDK 5 per C++ \[Download dell'SDK\] \[Guida di riferimento\]](#)
- [Server SDK 5 per C# \[Download dell'SDK\] \[Guida di riferimento\]](#)
- [Server SDK 5 for Go \[Download dell'SDK\] \[Guida di riferimento\]](#)

Crea il tuo server di gioco per Linux

Le flotte di GameLift container Amazon supportano server di gioco eseguiti su una piattaforma Linux. Ecco alcuni suggerimenti per creare un server di gioco per un target Linux:

- Se stai sviluppando il tuo gioco con il motore di gioco Unity, l'editor di gioco fornisce supporto integrato senza requisiti speciali per la creazione per Linux.
- Se stai sviluppando il tuo gioco in C++, devi includere le librerie OpenSSL per Linux quando crei l'Amazon Server GameLift SDK per C++ e quando crei il tuo server di gioco. Includi anche le stesse librerie nell'immagine del contenitore del server di gioco.
- Se stai sviluppando il tuo gioco con Unreal Engine su Windows, prendi in considerazione queste opzioni:
 - Collabora con Unreal Engine per configurare una catena di strumenti di [compilazione incrociata](#).
 - Configura uno spazio di lavoro Linux separato o utilizza uno strumento come il sottosistema Windows per Linux (WSL). Puoi usare questo ambiente per eseguire l'Unreal Editor su Linux per creare il tuo server di gioco.

Testa la tua integrazione localmente

Puoi testare l'integrazione del gioco a livello locale utilizzando una GameLift Anywhere flotta Amazon. Questo approccio è una best practice per aiutare a isolare i problemi direttamente correlati all'integrazione. Una Anywhere flotta è uno strumento utile per eseguire app di test e scenari di gioco, ad esempio avviare/interrompere sessioni di gioco e tracciare le connessioni dei giocatori. Puoi creare e testare in modo iterativo molto più velocemente con una Anywhere flotta, che offre una maggiore visibilità sulle attività di hosting.

Consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#) la pagina per ricevere assistenza sull'utilizzo di una GameLift Anywhere flotta Amazon per i test di integrazione. Il flusso di lavoro per la configurazione di un ambiente di test è simile al seguente:

1. Configura un dispositivo locale su cui è in esecuzione Linux.
2. Configura una Anywhere flotta. Crea una posizione personalizzata per il tuo dispositivo locale, crea una Anywhere flotta e quindi registra il tuo dispositivo locale come computer nel parco macchine.
3. Ottieni un token di autenticazione per il tuo server di gioco. Il processo server integrato richiede un token per l'autenticazione con il GameLift servizio Amazon. Puoi riutilizzare lo stesso token per

più processi server eseguiti contemporaneamente. Questo passaggio è necessario solo quando si utilizza una Anywhere flotta per i test di integrazione.

Note

I token di autenticazione sono temporanei e devono essere aggiornati regolarmente. Prendi in considerazione l'aggiunta di uno script al pacchetto di build del server per richiedere un nuovo token.

4. Aggiorna il codice del server di gioco per Anywhere. Quando è in esecuzione su una flotta Anywhere, il server di gioco deve chiamare il server SDK action `InitSdk()` ([C++](#)) ([C#](#)) ([Unreal](#)) con i seguenti parametri del server. Questo passaggio è necessario solo quando si utilizza una Anywhere flotta per i test di integrazione. Dopo aver aggiunto Amazon GameLift Agent all'immagine del contenitore, questi parametri vengono gestiti automaticamente.

Come best practice, configura il codice del server per estrarre questi valori dalle variabili di ambiente o dagli argomenti della console specificati all'avvio.

- `websocketUrl`— Utilizza il valore di `GameLiftServiceSdkEndpoint`, che viene restituito dalla chiamata `register-compute`.
 - `processId`— Assegna un identificatore univoco per il processo del server.
 - `fleetId`— L'identificatore della flotta Anywhere, che viene restituito dalla chiamata a `create-fleet`
 - `authToken`— Un token di autenticazione valido, che viene restituito dalla chiamata a `get-compute-auth-token`.
5. Sul computer locale, configura il server di gioco, crea il software e avvia un processo server.

Se l'integrazione del server ha esito positivo, il processo server richiama l'azione SDK del server `InitSDK()` per stabilire la connessione con il GameLift servizio Amazon, seguita da una chiamata per notificare `ProcessReady()` al servizio che è pronto per ospitare una sessione di gioco.

6. Inizia una sessione di gioco. Se hai integrato il client di gioco per richiedere una sessione di gioco, puoi utilizzarlo per richiedere una nuova sessione di gioco. In caso contrario, utilizzare il AWS comando CLI. [create-game-session](#) Amazon GameLift crea un `GameSession` oggetto e avvia il processo per avviare una nuova sessione di gioco.

Se la tua integrazione funziona, Amazon GameLift chiama un processo server sulla tua workstation locale per avviare una nuova sessione di gioco (usando il `onStartGameSession()` callback). Quando una sessione di gioco è pronta per i giocatori, il processo del server

chiama. `ActivateGameSession()` In risposta, Amazon GameLift aggiorna le informazioni `GameSession` sullo stato e sulla connessione in modo che un client di gioco possa connettersi alla sessione di gioco e giocare.

Prepara un'immagine del contenitore con il software del tuo server di gioco

Questa documentazione riguarda una funzionalità che è in versione di anteprima pubblica. ed è soggetta a modifiche.

Il container è l'elemento fondamentale di una flotta di GameLift container Amazon. Il contenitore include il server di gioco, insieme alle sue dipendenze come SDK, software, directory e file.

Per funzionare in una flotta di container, il server di gioco deve funzionare su Linux ed essere integrato con il server SDK 5.x.

Argomenti

- [Configura la tua directory di lavoro](#)
- [Crea l'immagine del tuo container](#)
- [Invia l'immagine del contenitore ad Amazon ECR](#)

Configura la tua directory di lavoro

La tua directory di lavoro è dove metti tutti i file necessari per creare l'immagine del contenitore e definire come Amazon la GameLift esegue.

Per configurare la directory di lavoro del contenitore

1. Crea la directory in cui desideri lavorare con le immagini dei tuoi GameLift container Amazon.

Example

Per esempio:

```
[~/]$ mkdir -p work/glc/gamebuild && cd work && find .  
.  
./glc  
./glc/gamebuild
```

2. Clona [Amazon GameLift Agent](#).

Example

Per esempio:

```
[~/work]$ git clone https://github.com/aws/amazon-gamelift-agent.git
Cloning into 'amazon-gamelift-agent'...
```

3. [Costruiscilo GameLiftAgent usando Maven](#).

Example

Per esempio:

```
[~/work]$ cd amazon-gamelift-agent
```

Example

```
[~/work/amazon-gamelift-agent]$ mvn clean compile assembly:single && \
mv target ../glc && cd .. && find glc
```

4. Aggiungi un server di gioco integrato con il server SDK 5.x, costruito e impacchettato in un file .ZIP
5. Copia il file in .ZIP ~/work/glc/gamebuild/

Se non disponi di un server di gioco SDK 5.x, puoi scaricare e utilizzare il nostro [SimpleServer](#) gioco di esempio per provare a utilizzare una flotta di container.

Example

```
[~/work]$ curl -o glc/gamebuild/SimpleServer.zip \
'https://ws-assets-prod-iad-r-iad-ed304a55c2ca1aee.s3.us-east-1.amazonaws.com/086bb355-4fdc-4e63-8ca7-af7cfc45d4f2/AmazonGameLiftSampleServerBinary.zip' &&
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload  Total    Spent    Left  Speed
100 5140k  100 5140k    0     0  12.3M    0  ---:---:--  ---:---:--  ---:---:-- 12.3M
glc
glc/target
```

```
glc/target/GameLiftAgent-1.0.jar
glc/gamebuild
glc/gamebuild/SimpleServer.zip
```

Crea l'immagine del tuo container

Il tuo Dockerfile specifica l'ambiente, il software e le istruzioni per creare il tuo contenitore.

Per creare il tuo Dockerfile

1. Spostati nella sottodirectory. `glc`

Example

```
[~/work]$ cd glc && find
.
./target
./target/GameLiftAgent-1.0.jar
./gamebuild
```

2. Crea e apri un nuovo Dockerfile.

Example

Per esempio:

```
[~/work/glc]$ nano Dockerfile
```

3. Copia da uno dei seguenti modelli, quindi incolla il contenuto nel tuo Dockerfile.

Modello di Dockerfile per il tuo server di gioco

Questo modello contiene le istruzioni minime di cui un container ha bisogno per essere utilizzabile in una GameLift flotta Amazon. Modifica il contenuto in base alle esigenze del tuo server di gioco.

```
# Base image
# -----
# Add the base image that you want to use over here,
# Make sure to use an image with the same architecture as the
# Instance type you are planning to use on your fleets.
# We require JDK to be installed in the base image, so that
```

```
# it can be used to run the &AGS; Agent
FROM public.ecr.aws/amazoncorretto/amazoncorretto:17-amd64
#
# Game build directory
# -----
# Add your game build to gamebuild directory and add the zip file name in the
'GAME_BUILD_ZIP' env variable below.
# The game build provided over here needs to be integrated with gamelift server sdk.
# This template assumes that the game build is in a zip format.
ENV GAME_BUILD_ZIP="<<ADD_GAME_BUILD_ZIP_FILE_NAME>" \
#
# Default directory
# -----
# Default directory, the value provided here should be where the game executable
exists.
# Provide this same value as your launch path in RuntimeConfiguration when creating a
fleet.
# Ref: https://docs.aws.amazon.com/gamelift/latest/apireference/
API\_ServerProcess.html
GAME_EXECUTABLE="<<ADD NAME OF EXECUTABLE WITHIN THE GAME BUILD>" \
HOME_DIR="/local/game" \
#
# Registered compute in anywhere fleet (not used in container fleets)
# -----
# Add the name for the registered compute in an anywhere fleet.
# This environment variable is required only for anywhere fleets, but not for
container fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
GAMELIFT_COMPUTE_NAME="<<ADD_COMPUTE_NAME>" \
#
# Default Gamelift Agent jar
# -----
GAMELIFT_AGENT_EXEC="GameLiftAgent-1.0.jar" \
#
# This env variable defines the name of the S3 bucket that stores the GameLift Agent
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
GAMELIFT_AGENT_LOGS_BUCKET_NAME="<<ADD NAME OF GAMELIFT AGENT LOGS S3 BUCKET>" \
#
# -----
```

```

# This env variable defines the name of the S3 bucket that stores the game session
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
# -----
GAME_SESSION_LOGS_BUCKET_NAME="<ADD NAME OF GAME SESSION LOGS S3 BUCKET>" \
#
# -----
GAMELIFT_AGENT_LOGS_PATH="/local/game/agentlogs/" \
#
# NOT USED in container fleets - USED in Anywhere fleets
# -----
# Specify the type of compute resource used to host the game servers.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
COMPUTE_TYPE="ANYWHERE" \
#
# Specify the credential to be used for creating the client.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
CREDENTIAL_PROVIDER="environment-variable"

USER root

# install dependencies as necessary
RUN yum install -y sudo \
    unzip \
    git \
    shadow-utils \
    iputils \
    tar \
    gcc \
    make \
    openssl-devel \
    zlib-devel \

```



```
vim \  
net-tools \  
nc \  
procps  
  
# Set up the ground for 'gamescale' user  
RUN groupadd -r gamescale -g 500 && \  
useradd -u 500 -r -g gamescale -m -s /sbin/nologin -c "Gamescale user" gamescale  
&& \  
echo "gamescale ALL=(ALL) NOPASSWD: ALL" | (EDITOR="tee -a" visudo) && \  
mkdir -p $HOME_DIR && \  
mkdir $HOME_DIR/mono && \  
chown -R gamescale:gamescale $HOME_DIR  
  
WORKDIR $HOME_DIR  
  
# extract game build as necessary  
COPY ./gamebuild/$GAME_BUILD_ZIP .  
RUN unzip ./GAME_BUILD_ZIP -d ./  
  
# copy Gamelift Agent jar  
COPY ./gameliftAgent/$GAMELIFT_AGENT_EXEC ./  
  
# Add permissions to game build and gamelift agent jar  
RUN chmod +x ./GAME_EXECUTABLE  
RUN chmod +x ./GAMELIFT_AGENT_EXEC  
  
# Check if java is installed on the image, if not then the Agent will not be able  
to run  
RUN java --version  
  
USER gamescale  
  
ENV PATH="$PATH:$HOME_DIR/bin:$JAVA_HOME"  
  
# Change directory to bin  
WORKDIR $HOME_DIR  
  
# check path before starting the container  
RUN echo $PATH  
  
# Create logs directory for GameLift Agent & server processes  
RUN mkdir logs  
RUN mkdir agentlogs
```

```

# Start the GameLift Agent
ENTRYPOINT sleep 90 && java -jar $GAMELIFT_AGENT_EXEC -ip "192.168.1.1" -gs1b
"$GAME_SESSION_LOGS_BUCKET_NAME" -galb "$GAMELIFT_AGENT_LOGS_BUCKET_NAME" -galp
"$GAMELIFT_AGENT_LOGS_PATH" -glc environment-variable

```

Dockerfile per l'esempio **SimpleServer**

```

# Base image
# -----
# Add the base image that you want to use over here,
# Make sure to use an image with the same architecture as the
# Instance type you are planning to use on your fleets.
# We require JDK to be installed in the base image, so that
# it can be used to run the &AGS; Agent
FROM public.ecr.aws/amazoncorretto/amazoncorretto:17-amd64
#
# Game build directory
# -----
# Add your game build to gamebuild directory and add the zip file name in the
'GAME_BUILD_ZIP' env variable below.
# The game build provided over here needs to be integrated with gamelift server sdk.
# This template assumes that the game build is in a zip format.
ENV GAME_BUILD_ZIP="SimpleServer.zip" \
#
# Default directory
# -----
# Default directory, the value provided here should be where the game executable
exists.
# Provide this same value as your launch path in RuntimeConfiguration when creating a
fleet.
# Ref: https://docs.aws.amazon.com/gamelift/latest/apireference/
API\_ServerProcess.html
GAME_EXECUTABLE="GameLiftSampleServer" \
HOME_DIR="/local/game" \
#
# Registered compute in anywhere fleet (not used in container fleets)
# -----
# Add the name for the registered compute in an anywhere fleet.
# This environment variable is required only for anywhere fleets, but not for
container fleets.
# If it is set for container fleets, it will be overridden by Gamelift.

```

```
GAMELIFT_COMPUTE_NAME="<ADD_COMPUTE_NAME>" \  
#  
# Default Gamelift Agent jar  
# -----  
GAMELIFT_AGENT_EXEC="GameLiftAgent-1.0.jar" \  
#  
# This env variable defines the name of the S3 bucket that stores the GameLift Agent  
logs.  
# This S3 bucket should exist in the customer AWS account.  
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would  
need to  
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during  
CreateFleet operation.  
GAMELIFT_AGENT_LOGS_BUCKET_NAME="<ADD NAME OF GAMELIFT AGENT LOGS S3 BUCKET>" \  
#  
# -----  
# This env variable defines the name of the S3 bucket that stores the game session  
logs.  
# This S3 bucket should exist in the customer AWS account.  
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would  
need to  
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during  
CreateFleet operation.  
# -----  
GAME_SESSION_LOGS_BUCKET_NAME="<ADD NAME OF GAME SESSION LOGS S3 BUCKET>" \  
#  
# -----  
GAMELIFT_AGENT_LOGS_PATH="/local/game/agentlogs/" \  
#  
# NOT USED in container fleets - USED in Anywhere fleets  
# -----  
# Specify the type of compute resource used to host the game servers.  
# This env variable is required only for anywhere fleets, but not for container  
fleets.  
# If it is set for container fleets, it will be overridden by Gamelift.  
#  
# -----  
COMPUTE_TYPE="ANYWHERE" \  
#  
# Specify the credential to be used for creating the client.  
# This env variable is required only for anywhere fleets, but not for container  
fleets.  
# If it is set for container fleets, it will be overridden by Gamelift.  
#
```

```
# -----
CREDENTIAL_PROVIDER="environment-variable"

USER root

# intall dependencies as necessary
RUN yum install -y sudo \
    unzip \
    git \
    shadow-utils \
    iputils \
    tar \
    gcc \
    make \
    openssl-devel \
    zlib-devel \
    vim \
    net-tools \
    nc \
    procps

# Set up the ground for 'gamescale' user
RUN groupadd -r gamescale -g 500 && \
    useradd -u 500 -r -g gamescale -m -s /sbin/nologin -c "Gamescale user" gamescale
&& \
    echo "gamescale ALL=(ALL) NOPASSWD: ALL" | (EDITOR="tee -a" visudo) && \
    mkdir -p $HOME_DIR && \
    mkdir $HOME_DIR/mono && \
    chown -R gamescale:gamescale $HOME_DIR

WORKDIR $HOME_DIR

# extract game build as necessary
COPY ./gamebuild/$GAME_BUILD_ZIP .
RUN unzip ./ $GAME_BUILD_ZIP -d ./

# copy Gamelift Agent jar
COPY ./target/$GAMELIFT_AGENT_EXEC ./

# Add permissions to game build and gamelift agent jar
RUN chmod +x ./ $GAME_EXECUTABLE
RUN chmod +x ./ $GAMELIFT_AGENT_EXEC
```

```
# Check if java is installed on the image, if not then the Agent will not be able
to run
RUN java --version

USER gamescale

ENV PATH "$PATH:$HOME_DIR/bin:$JAVA_HOME"

# Change directory to bin
WORKDIR $HOME_DIR

# check path before starting the container
RUN echo $PATH

# Create logs directory for GameLift Agent & server processes
RUN mkdir logs
RUN mkdir agentlogs

# Start the GameLift Agent
ENTRYPOINT sleep 90 && java -jar $GAMELIFT_AGENT_EXEC -ip "192.168.1.1" -gslb
"$GAME_SESSION_LOGS_BUCKET_NAME" -galb "$GAMELIFT_AGENT_LOGS_BUCKET_NAME" -galp
"$GAMELIFT_AGENT_LOGS_PATH" -glc environment-variable
```

Note

Nota: alcune variabili di ambiente nel Dockerfile possono essere sovrascritte da.

[ContainerDefinition](#)

Per creare l'immagine del contenitore

1. Crea l'immagine del tuo contenitore.

Se utilizzi il tuo server SDK 5.x

Puoi specificare il nome del repository locale che desideri.

Example

```
[~/work/glc]$ docker build -t <local repository name>:<optional tag> .
```

Se stai usando il nostro esempio **SimpleServer**

Example

```
[~/work/glc]$ docker build -t simple-server:version-1 .  
Successfully built 0123456789012  
Successfully tagged simple-server:version-1
```

Note

Negli esempi seguenti, utilizziamo *simpler-server* come REPOSITORY valore iniziale e *version-1* come valore. TAG

2. Visualizza l'elenco delle immagini e prendi nota dei valori REPOSITORY and IMAGE ID. Ti serviranno nella procedura riportata di seguito.

Example

```
[~/work/glc]$ docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE  
simple-server        version-1    0123456789012    14 minutes ago  1.24GB
```

Invia l'immagine del contenitore ad Amazon ECR

Carica l'immagine del contenitore in un repository privato in Amazon ECR. Quando crei una definizione di gruppo di container, fai riferimento a questa posizione del repository in modo che Amazon GameLift possa scattare un'istantanea dell'immagine del tuo container e utilizzarla per distribuire una flotta di container.

Note

Se non disponi ancora di un repository privato Amazon ECR, [creane](#) uno.

Per ottenere le tue credenziali Amazon ECR

- Prima di poter inviare l'immagine del contenitore ad Amazon ECR, devi acquisire le AWS credenziali in formato temporaneo e fornirle a Docker. Ottieni le tue credenziali Amazon ECR per consentire a Docker di accedere.

Example

```
[~/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
```

Per inviare l'immagine del contenitore ad Amazon ECR

1. Copia l'URI del [repository privato Amazon ECR](#) che desideri utilizzare.
2. 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>
```

3. Invia l'immagine del contenitore ad Amazon ECR

Example

```
[~/work/glc]$ docker image push <Amazon ECR private repository URI>
```

Progetta una flotta di GameLift container Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Questi argomenti presentano le decisioni chiave che prenderai durante la configurazione di una flotta di GameLift container Amazon. Le tue decisioni influiscono sul modo in cui configuri le impostazioni per container, gruppi di container e flotte.

Argomenti

- [Progetta la struttura dei container della tua flotta](#)
- [Imposta i limiti delle risorse](#)
- [Designare contenitori essenziali](#)
- [Configurare le connessioni di rete](#)
- [Imposta i controlli sanitari per i container](#)
- [Imposta le dipendenze del contenitore](#)
- [Configura una flotta di container](#)

Progetta la struttura dei container della tua flotta

Come primo passo, identifica il software e le risorse necessari per ospitare il tuo server di gioco, tra cui:

- La tua applicazione per server di gioco. L'applicazione deve essere integrata con le GameLift funzionalità di Amazon per l'hosting, inclusa la versione SDK del server 5+. Per informazioni, consulta [Integra il tuo gioco con Amazon GameLift](#).
- L' GameLift agente Amazon. Questo agente on-computing mantiene la comunicazione con il GameLift servizio Amazon e gestisce il ciclo di vita di tutti i processi del server di gioco. Per ulteriori dettagli, consulta [Server di gioco e Amazon GameLift Agent](#).
- Software e risorse aggiuntivi, se necessario. Ciò potrebbe includere il software necessario per eseguire le applicazioni dei server di gioco. Il software di supporto comune viene utilizzato per la registrazione e il monitoraggio, la sicurezza, la distribuzione dei contenuti e la sincronizzazione dei dati.

Successivamente, decidi come strutturare il software e le risorse per una flotta di GameLift container Amazon. Amazon GameLift utilizza gruppi di contenitori per organizzare i contenitori. Una flotta ha sempre un gruppo di container di repliche e, facoltativamente, può avere una flotta di container daemon. Per ulteriori dettagli, consulta [Componenti della flotta di container](#).

- Inizia progettando il tuo gruppo di contenitori di repliche. Considera le linee guida seguenti:

- Raggruppa l'applicazione del server di gioco e Amazon GameLift Agent nello stesso contenitore. Rendi questo contenitore l'unico contenitore essenziale del gruppo di repliche.
- Organizza tutti gli altri software per il tuo server di gioco in contenitori. Potresti scegliere di mettere tutto in un unico contenitore nel gruppo di repliche. Oppure puoi scegliere di creare uno o più contenitori sidecar. Alcuni motivi per utilizzare i sidecar includono:
 - Per impostare una sequenza di avvio/spegnimento per singoli software. È possibile ottenere ciò inserendo il software in contenitori separati e impostando le dipendenze tra di essi.
 - Per impostare limiti specifici del contenitore per l'utilizzo della memoria e della CPU.
 - Per specificare diverse impostazioni di configurazione del contenitore per ogni contenitore, ad esempio un comando di avvio, un punto di ingresso, una directory di lavoro, variabili di ambiente o controlli di integrità.
- Decidi se hai bisogno di un gruppo di container daemon per la tua flotta. Considera i seguenti aspetti:
 - I contenitori daemon vengono in genere utilizzati per eseguire processi in background o di monitoraggio.
 - I contenitori in un gruppo di daemon non vengono replicati su un'istanza della flotta. Ciò significa che i contenitori di un gruppo di daemon non sono scalabili insieme al gruppo di contenitori di replica.
 - Un gruppo di daemon può avere più contenitori. È possibile designare come essenziale qualsiasi contenitore in un gruppo di daemon.

Imposta i limiti delle risorse

Per ogni gruppo di contenitori, determina la quantità di memoria e CPU di cui il gruppo ha bisogno per eseguire il software. Amazon GameLift si affida a queste informazioni per gestire le risorse per il gruppo di contenitori. Inoltre, utilizza queste informazioni per calcolare il numero di gruppi di container di repliche che un'immagine della flotta può contenere. È inoltre possibile impostare limiti per singoli contenitori.

Imposta limiti opzionali per i 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.

- `SoftLimit(memoria)` — Riserva una quantità minima di memoria 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.
- `HardLimit(memoria)` — Imposta un limite massimo di memoria per il contenitore. Se il contenitore supera questo limite, si verifica un riavvio.
- `Cpulimite`: riserva una quantità minima di risorse della CPU 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

Indica ad Amazon di GameLift quanta memoria e risorse CPU sono necessarie per ogni gruppo di container. L'obiettivo è allocare risorse sufficienti per ottimizzare le prestazioni dei server di gioco. Amazon GameLift utilizza questi limiti per calcolare come imballare gruppi di container di replica su un'istanza di flotta. Li utilizzerai anche per scegliere un tipo di istanza per una flotta di container.

Calcola la memoria totale e la CPU necessarie per tutti i processi in ogni contenitore di un gruppo. Considera i seguenti aspetti:

- Quali processi vengono eseguiti in tutti i contenitori del gruppo di contenitori? Somma le risorse necessarie per questi processi.
- Quanti processi simultanei del server di gioco intendi eseguire in ciascun gruppo di container? Hai impostato questo valore come parte della configurazione di runtime di una flotta, ma devi pianificare una quantità di memoria sufficiente per utilizzarli qui (vedi [Ottimizzazione della configurazione di runtime](#)).

In base alla stima dei requisiti del gruppo di contenitori, imposta le seguenti `ContainerGroupDefinition` proprietà:

- `TotalMemoryLimit`— 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 alla somma di tutti i limiti di memoria software del contenitore

- uguale o superiore al limite massimo di memoria rigida per un contenitore del gruppo
- `TotalCpuLimit` — Imposta un limite massimo di CPU 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 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 replica con i seguenti tre contenitori:

- Il contenitore A è il nostro contenitore di replica essenziale. Esegue i processi del server di gioco e Amazon GameLift Agent. Stimiamo i requisiti di risorse per un server di gioco a 512 MiB e 1024 CPU. Prevediamo di far eseguire al container 10 processi del server. Poiché questo contenitore esegue il nostro software più importante, abbiamo impostato una riserva di memoria software di 6144 MiB e nessun limite di memoria rigida o limite di riserva della CPU.
- Container B esegue software di supporto con requisiti di risorse stimati in 1024 MiB e 1536 CPU. Abbiamo impostato un limite di riserva di memoria software di 1024 MiB, un limite di memoria rigida di 2048 MiB e un limite di riserva della CPU di 1024 CPU.
- Il contenitore C esegue la registrazione non critica e altre utilità di monitoraggio. 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 (1) la somma dei limiti di memoria software (6144+1024 MiB) e (2) il limite massimo di memoria rigida (1024 MiB).
- Limite totale della CPU: 13312 CPU. Questo valore supera la somma del limite della CPU (1024+512 CPU).

Designare contenitori essenziali

Per ogni contenitore, designa il contenitore come essenziale o non essenziale. Tutti i gruppi di contenitori devono avere almeno un contenitore essenziale. Il contenitore essenziale svolge il lavoro fondamentale del gruppo di contenitori, ad esempio ospita i server di gioco. Si prevede sempre che il contenitore essenziale sia in esecuzione. Se fallisce, l'intero gruppo di contenitori si riavvia.

- Il gruppo di container di replica della tua flotta può avere esattamente un container essenziale. Questo contenitore esegue Amazon GameLift Agent e i processi del server di gioco che gestisce.

- Se la tua flotta ha un gruppo di container daemon, puoi designare più container essenziali. Rendi essenziale un contenitore daemon se desideri che un errore del contenitore richieda il riavvio di un gruppo di contenitori.

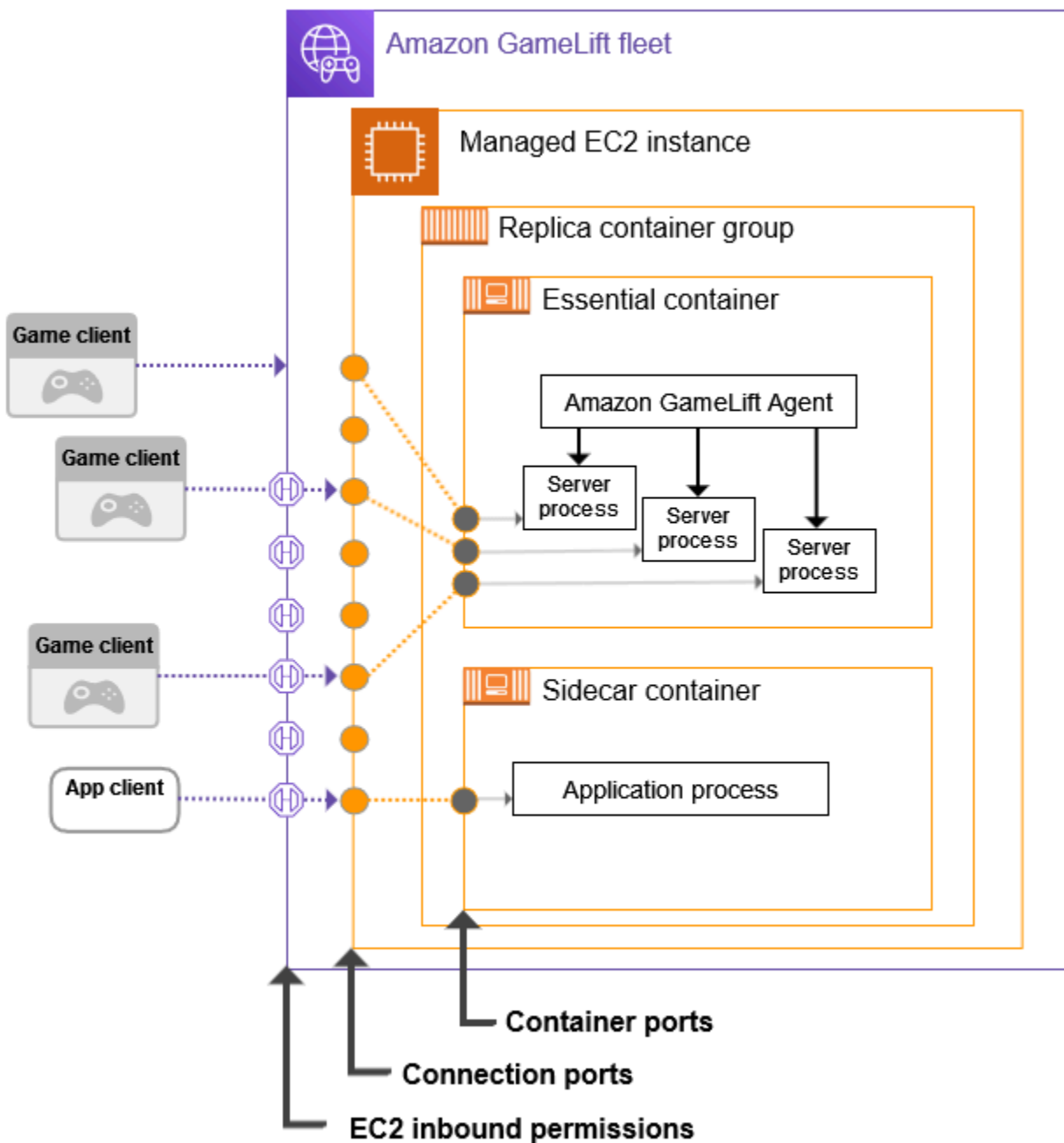
Imposta la `ContainerDefinition` proprietà `Essential` su `true` o `false` per ogni contenitore.

Configurare le connessioni di rete

È possibile stabilire 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 mantiene le mappature tra le porte interne dei container 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 fornisce 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. Non puoi modificare le configurazioni dei porti di una flotta esistente, ma puoi consentire o limitare l'accesso in base alle esigenze modificando le autorizzazioni in entrata. Ad esempio, puoi rimuovere le autorizzazioni per tutte le porte di connessione per impedire l'accesso ai container della flotta.



Imposta gli intervalli di porte del contenitore

Configura una definizione di contenitore con un numero sufficiente di porte container per qualsiasi processo che richieda l'accesso esterno. Alcuni contenitori non avranno bisogno di porte. Altri devono disporre di un numero sufficiente di porte per assegnarne una a ogni processo che ne necessita.

Il tuo gruppo di contenitori di repliche essenziale, che gestisce i tuoi server di gioco, necessita di una porta per ogni processo del server di gioco in esecuzione contemporaneamente (come configurato nella flotta). RuntimeConfiguration Il processo del server di gioco ascolta sulla porta assegnata e lo segnala ad Amazon GameLift.

Quando crei una definizione di gruppo di contenitori, definisci un intervallo di porte container per ogni contenitore che necessita dell'accesso alla rete (vedi [ContainerDefinitionInput: PortConfiguration](#)). Assicurati che l'intervallo sia sufficientemente ampio da assegnare una porta a ogni processo che ne necessita. Ai processi devono essere assegnati numeri di porta nella configurazione delle porte del contenitore.

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 assegna le porte di connessione e le mappa alle porte dei container in base alle esigenze.

Quando crei una flotta di container, definisci un intervallo di porte di connessione (vedi [ContainerGroupsConfiguration: ConnectionPortRange](#)). Assicurati che l'intervallo contenga un numero sufficiente di porte da mappare su ogni porto container di un'istanza della flotta. Per calcolare le porte di connessione minime necessarie, utilizza la seguente formula:

```
[Total number of container ports defined for containers in the replica container group] * [Number of replica container groups per instance] + [Total number of container ports defined for containers in the daemon 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 replica per istanza. Se una flotta dispone di porte di connessione sufficienti per un gruppo di contenitori di replica per istanza, Amazon GameLift implementerà solo un gruppo di contenitori di replica, anche se le istanze dispongono di potenza di calcolo sufficiente per più gruppi di contenitori di repliche.

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.

[Quando crei una flotta di container, definisci un set di autorizzazioni in entrata \(vedi:EC2\). CreateFleet InboundPermissions](#) Imposta le proprietà delle porte di autorizzazione in entrata per includere alcuni o tutti i valori nelle impostazioni delle porte di connessione della flotta. Per modificare le autorizzazioni in entrata su una flotta di container esistente, chiama. [UpdateFleetPortSettings](#)

Scenario di esempio

Questo esempio illustra come impostare tutte e tre le proprietà della connessione di rete.

- Il gruppo di container di replica della nostra flotta ha 1 container, che esegue i processi del server di gioco. La configurazione del runtime indica al contenitore di eseguire 10 processi simultanei del server di gioco.

Nella definizione del gruppo di contenitori di replica, 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 daemon con 1 container. Ha 1 processo che richiede l'accesso alla rete. Nella definizione del gruppo di contenitori daemon, impostiamo il `PortConfiguration` parametro per questo contenitore come segue:

```
"PortConfiguration": {
  "ContainerPortRanges": [ { "FromPort": 25, "ToPort": 25, "Protocol": "TCP" } ] }
```

- La nostra flotta è configurata con 3 gruppi di container di replica per istanza di flotta. Alla luce di queste informazioni, possiamo utilizzare la formula per calcolare il numero di porte di connessione di cui abbiamo bisogno:
 - Minimo: 31 porte [10 porte container di replica * 3 gruppi di contenitori di replica per istanza + 1 porta contenitore daemon]
 - Procedura ottimale: 62 porte [porte minime* 2]

Durante la creazione della flotta di container, impostiamo il `ConnectionPortRange` parametro `ContainerGroupsConfiguration` 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 `EC2InboundPermissions` parametro come segue:

```
"EC2InboundPermissions": [  
  {"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 il contenitore è essenziale, l'intero gruppo di contenitori si riavvia.

È possibile definire criteri personalizzati aggiuntivi per misurare lo stato del contenitore 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 i controlli sanitari opzionali in base al tipo di contenitore come segue:

- Per un contenitore di replica essenziale, non configurate i controlli di integrità. Amazon GameLift Agent gestisce automaticamente i report sullo stato di salute di questo contenitore.
- Per i contenitori di replica non essenziali e per qualsiasi contenitore daemon, puoi facoltativamente impostare i parametri di controllo dello stato.

Imposta le seguenti `ContainerDefinition` proprietà per il controllo dello stato di un 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, prendi in considerazione 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 Regione AWS GameLift supportato da Amazon. 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, consulta [Sedi GameLift di hosting Amazon](#).

Scegli un tipo e una dimensione di istanza per il tuo parco istanze

Amazon GameLift supporta un'ampia gamma di tipi di istanze Amazon EC2, tutte disponibili per l'uso con una flotta di container. La disponibilità e il prezzo del tipo di istanza variano in base alla località. Puoi visualizzare un elenco dei tipi di istanze supportati, filtrati per posizione, nella GameLift console Amazon (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 istanze EC2](#). All'interno di ogni famiglia hai una gamma di 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, è preferibile scegliere una dimensione che consenta di ospitare più copie del gruppo di contenitori di replica 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.

Ottimizza la configurazione del runtime

La configurazione di runtime di una flotta è un insieme di istruzioni su come eseguire i processi del server per l'hosting di sessioni di gioco. Queste istruzioni sono implementate dall' GameLift agente Amazon in ogni gruppo di container di replica della flotta.

La configurazione di runtime di una flotta determina il numero di processi server eseguiti contemporaneamente in ciascun gruppo di contenitori di repliche. Questa impostazione influisce sul modo in cui si calcolano i limiti di risorse del gruppo di container e sulla scelta del tipo di istanza per la flotta. È necessario bilanciare questi tre elementi durante la progettazione della flotta.

Per ulteriori informazioni su come utilizzare le configurazioni di runtime, consulta [Gestisci il modo in cui Amazon GameLift lancia i server di gioco](#).

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. Per informazioni, consulta [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.

Crea definizioni di gruppi di container per una flotta di GameLift container Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

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 identifica il set di container da far funzionare sulla flotta e come eseguirli. Quando crei una flotta di container, specifichi le definizioni dei gruppi di container da distribuire nella flotta. Per ulteriori informazioni sui gruppi di container, consulta [Componenti della flotta di container](#).

Prima di iniziare

Completa le seguenti operazioni:

- Progetta un'architettura di container per ospitare i tuoi server di gioco. Per informazioni, consulta [Progetta una flotta di GameLift container Amazon](#).
- Pianifica le definizioni dei contenitori da includere nel gruppo di contenitori. Se utilizzi la AWS CLI, crea la definizione del contenitore in un file JSON.
- Invia le immagini finali del contenitore a un registro Amazon Elastic Container Registry (Amazon ECR) Regione AWS nello stesso in cui intendi creare il gruppo di contenitori. Amazon GameLift archivia un'istantanea di ogni immagine al momento della creazione della definizione del gruppo di

container e utilizza la copia per la distribuzione su una flotta di container. Per informazioni, consulta [Prepara un'immagine del contenitore con il software del tuo server di gioco](#).

- Verifica che il tuo AWS utente disponga delle autorizzazioni IAM per accedere al repository Amazon ECR. Per informazioni, consulta [Gestisci le autorizzazioni degli utenti per Amazon GameLift](#). Sono necessarie almeno le autorizzazioni per le seguenti azioni:
 - `ecr:DescribeImages`
 - `ecr:BatchGetImage`
 - `ecr:GetDownloadUrlForLayer`

Clonare una definizione di gruppo di contenitori

Puoi utilizzare la GameLift console Amazon per clonare una definizione di gruppo di contenitori esistente.

Per clonare un gruppo di contenitori

1. Nella [GameLift console Amazon](#), 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.

Crea una definizione di gruppo di contenitori di replica

Un gruppo di contenitori di repliche gestisce il software del server di gioco. Un gruppo di contenitori di replica ha almeno un contenitore che esegue Amazon GameLift Agent e i processi del server di gioco. Il gruppo potrebbe disporre di contenitori «secondari» aggiuntivi per eseguire il software di supporto.

Questo argomento descrive come creare una definizione di gruppo di contenitori utilizzando la GameLift console Amazon o gli strumenti AWS CLI. Per informazioni più dettagliate sull'impostazione

delle configurazioni dei gruppi di contenitori, consulta [Progetta una flotta di GameLift container Amazon](#)

Console

Nella [GameLift console Amazon](#), seleziona il Regione AWS punto in cui desideri creare il gruppo di contenitori.

Apri la barra di navigazione a sinistra della console e scegli Gruppi di contenitori. Nella pagina Gruppi di contenitori, scegli Crea gruppo di contenitori.

Fase 1: Definire i dettagli del gruppo.

1. Immettere un nome per la definizione del gruppo di contenitori. Questo nome deve essere univoco per la regione Account AWS and. Nella console, le definizioni dei gruppi sono elencate per nome, quindi può essere utile assegnare etichette significative.
2. Seleziona la strategia di pianificazione delle repliche.
3. Per Limite di memoria totale, inserisci la memoria massima disponibile per il gruppo di contenitori. Per informazioni sul calcolo di questo valore, consulta [Imposta i limiti delle risorse](#).
4. Per Limite totale della CPU, inserisci la potenza di calcolo massima disponibile per il gruppo di contenitori. Per informazioni sul calcolo di questo valore, consulta [Imposta i limiti delle risorse](#).

Fase 2: Aggiungere le definizioni dei contenitori.

Definisci il contenitore con la tua applicazione server di gioco e Amazon GameLift Agent. Questo è il tuo contenitore di replica essenziale.

1. Fornisci un nome di definizione del contenitore. Ogni contenitore definito per il gruppo deve avere un valore di nome univoco.
2. Identifica l'URI dell'immagine Amazon ECR dell'immagine del contenitore. Inserisci 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. Per Essential container, Yes viene selezionato automaticamente per la prima definizione del contenitore. Se aggiungi un'altra definizione di contenitore, puoi attivare o disattivare questa impostazione per ogni definizione. Per ulteriori dettagli, consulta [Designare contenitori essenziali](#).
 4. Imposta uno o più intervalli di porte interne del contenitore. Questo contenitore ospita i tuoi server di gioco, quindi definisci un intervallo con abbastanza porte per ogni processo del server da eseguire nel gruppo di contenitori. Per ulteriori dettagli, consulta [Configurare le connessioni di rete](#).
 5. Le impostazioni opzionali Overrides e Environment variabili consentono di specificare i valori da passare al contenitore all'avvio. I valori impostati qui sostituiscono tutte le impostazioni già presenti nell'immagine del contenitore.
 6. Imposta limiti opzionali per il contenitore per gestire l'allocazione delle risorse per questo contenitore. Per ulteriori dettagli, consulta [Imposta i limiti delle risorse](#).
 7. Definisci contenitori aggiuntivi non essenziali secondo necessità:
 - Fornisci un nome di definizione del contenitore e l'URI dell'immagine ECR. I contenitori non essenziali non devono eseguire Amazon GameLift Agent.
 - Imposta un intervallo di porte interne per i container solo se i contenitori hanno processi che richiedono l'accesso alla rete.
 - Opzionalmente, imposta un Health check per il contenitore. Quando un contenitore non essenziale non supera un controllo di integrità, richiede solo il riavvio del contenitore in errore.
 - Se necessario, imposta le sostituzioni, le variabili di ambiente e i limiti di allocazione delle risorse.

Fase 3: Configurare le dipendenze.

Se nella definizione del gruppo di contenitori è presente più di un contenitore, è possibile definire le dipendenze tra di essi. Utilizza le dipendenze per configurare le sequenze di avvio e spegnimento in base alle condizioni del contenitore. Per ulteriori dettagli, consulta [Imposta le dipendenze del contenitore](#).

1. Identifica il nome del contenitore per cui desideri aggiungere una dipendenza. Questo contenitore non si avvia finché non viene soddisfatta la condizione di dipendenza.
2. Identifica la dipendenza Nome e condizione del contenitore. Questo contenitore deve soddisfare la condizione prima che il contenitore dipendente possa iniziare.
3. Imposta dipendenze aggiuntive secondo necessità. Puoi creare più dipendenze per qualsiasi contenitore. Evita di creare dipendenze circolari.

Fase 4: Rivedi e crea.

1. Controlla tutte le impostazioni di definizione del gruppo di contenitori. Non è possibile modificare la configurazione di una definizione di gruppo di contenitori dopo la sua creazione. Usa Modifica per apportare modifiche a qualsiasi sezione, inclusa ciascuna definizione di contenitore 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`, quando Amazon GameLift inizia a scattare istantanee di tutte le immagini dei container 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.

Example : gruppo di contenitori Replica

Questo esempio illustra una richiesta per una definizione di gruppo di contenitori di replica. La struttura dei comandi per la creazione di definizioni di repliche e gruppi di daemon è essenzialmente identica. I dettagli specifici per ogni tipo di gruppo sono descritti nelle definizioni dei singoli contenitori.

Questo esempio 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 \  
  --scheduling-strategy REPLICA \  
  --total-memory-limit 4096 \  
  --total-cpu-limit 1024 \  
  --container-definitions file://SimpleServer.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](#) Amazon GameLift API Reference.
3. Salvate il file localmente in modo da potervi fare riferimento in un comando AWS CLI.

Esempio: definizione essenziale del contenitore di replica

Example

Questo esempio descrive il contenitore essenziale per il gruppo di contenitori di replica. Il contenitore di replica essenziale include l'applicazione per server di gioco, Amazon GameLift Agent, 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": "SimpleServer",
    "ImageUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/gl-containers:complex-server",
    "Essential": true,
    "Cpu": 256,
    "MemoryLimits": {
      "HardLimit": 128
    },
    "PortConfiguration": {
      "ContainerPortRanges": [
        {
          "FromPort": 2000,
          "Protocol": "TCP",
          "ToPort": 2100
        }
      ]
    }
  }
]
```

Crea una flotta di GameLift container Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Dopo aver creato le definizioni dei gruppi di container, usa la [GameLift console Amazon](#) o il AWS Command Line Interface (AWS CLI) per creare una flotta di container.

Dopo aver creato una nuova flotta, lo stato della flotta passa attraverso diverse fasi man mano che Amazon GameLift distribuisce i gruppi di container su ciascuna istanza della flotta e avvia i server

di gioco. Quando la flotta raggiunge lo stato ACTIVE, è pronta per ospitare sessioni di gioco. Per assistenza in caso di problemi con la creazione del parco istanze, consulta [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#).

Console

Nella [GameLift console Amazon](#), seleziona il Regione AWS luogo in cui desideri creare la flotta. Le definizioni dei gruppi di container devono trovarsi nella stessa regione in cui desideri creare la flotta.

Apri la barra di navigazione a sinistra della console e scegli Flotte. Nella pagina Flotte, scegli Crea flotta.

Passaggio 1: scegli il tipo di elaborazione

- Scegli il tipo di calcolo Containers.

Fase 2: Definizione dei dettagli della flotta

1. Nella sezione Dettagli della flotta, inserisci il nome e la descrizione della flotta.
2. Nella sezione Dettagli del gruppo di container, identifica i gruppi di container da distribuire nella flotta. È necessario aggiungere un gruppo di contenitori di replica. Facoltativamente, puoi aggiungere un gruppo di contenitori daemon. Ogni gruppo deve essere in stato. READY
3. Imposta l'intervallo di porte di connessione per la flotta. Per ulteriori dettagli, consulta [Configurare le connessioni di rete](#).
4. Facoltativamente, specifica le repliche desiderate per istanza da distribuire. Puoi specificare il numero desiderato o lasciare che Amazon GameLift calcoli il numero massimo possibile. Se specifichi un numero desiderato maggiore del massimo calcolato, la creazione della flotta avrà esito negativo. Non puoi modificare questa impostazione dopo la creazione del parco veicoli. Per maggiori dettagli sull'imballaggio di gruppi di contenitori di repliche, consulta [Concetti principali](#).
5. (Facoltativo) In Dettagli aggiuntivi:
 - a. Ad esempio, specifica un ruolo IAM che autorizzi le applicazioni della build del gioco ad accedere ad altre AWS risorse del tuo account. Per ulteriori informazioni, consulta [Comunica con altre AWS risorse delle tue flotte](#). Per creare una flotta con un ruolo di istanza, il tuo account deve disporre dell'PassRole autorizzazione IAM. Per ulteriori informazioni, consulta [Esempi di autorizzazioni IAM per Amazon GameLift](#).

- b. Per Metric group, inserisci il nome di un gruppo metrico della flotta nuovo o esistente. Puoi aggregare le metriche per più flotte aggiungendole allo stesso gruppo di metriche.

Fase 3: 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 *Gestire in. Regioni AWS* \[Riferimenti generali di AWS\]\(#\)](#)

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, consulta [Configura una flotta di container](#). La dimensione del tipo di istanza scelto influirà sul modo in cui i gruppi di container di repliche vengono impacchettati in ciascuna istanza del parco istanze. A seconda della scelta, valuta la possibilità di rivedere le impostazioni per le repliche desiderate per istanza.

Fase 4: Configurazione del runtime

La configurazione di runtime determina il modo in cui i processi del server di gioco vengono avviati ed eseguiti. Queste istruzioni vengono passate ad Amazon GameLift Agent, che le implementa allo stesso modo in ogni gruppo di contenitori di replica. Puoi aggiornare la configurazione di runtime di una flotta chiamando [UpdateRuntimeConfiguration](#).

1. Per Launch path, inserisci il percorso di un file eseguibile di gioco.
2. (Facoltativo) Per i parametri di avvio, inserisci le informazioni da passare all'eseguibile del gioco come set di parametri della riga di comando.
3. Specificate il numero di processi simultanei da mantenere in esecuzione in ogni gruppo di contenitori di repliche. Controlla le GameLift [quote](#) Amazon sul numero di processi server per istanza. Al totale dei processi simultanei, si applicano dei limiti sui processi simultanei del server per istanza per tutte le configurazioni. Se configuri la flotta in modo da superare il limite, la flotta non può attivarsi.
4. Imposta limiti opzionali per le attivazioni simultanee delle sessioni di gioco. Queste impostazioni consentono di limitare la quantità di risorse consumate all'avvio di una nuova sessione di gioco. Le attivazioni delle sessioni di gioco possono avere un impatto sulle prestazioni delle sessioni di gioco esistenti.
5. Imposta le impostazioni delle porte EC2 per consentire al traffico esterno di accedere ai processi in esecuzione sulla flotta. Specificate alcuni o tutti i numeri delle porte di connessione definiti per il parco veicoli. Non devi impostare queste porte quando crei la flotta, ma senza di esse nessun traffico può connettersi ai tuoi server di gioco. Per aggiornare le impostazioni dei porti di una flotta in un secondo momento, chiama [UpdateFleetPortSettings](#).
6. In Impostazioni delle risorse della sessione di gioco, configura le seguenti funzionalità opzionali:
 - a. Attiva o disattiva la politica di protezione di Game Scaling. Con la protezione attiva, Amazon GameLift non chiuderà le istanze durante un evento di scalabilità se ospitano una sessione di gioco attiva.
 - b. Imposta un limite massimo per la creazione di risorse per limitare il numero di sessioni di gioco che un giocatore può creare in un periodo di tempo specificato.

Fase 5: Configurare i tag

- (Facoltativo) Aggiungi tag alla build inserendo le coppie Chiave e Valore. Scegli Avanti per continuare con la revisione della creazione della flotta.

Fase 6: Rivedi e crea.

- 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 [Gestisci le tue GameLift flotte Amazon](#). Puoi aggiornare la capacità della flotta dopo che la flotta ha raggiunto lo stato ATTIVO. Per ulteriori informazioni, consulta [Scalabilità della capacità di GameLift hosting di Amazon](#). 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, quando Amazon GameLift 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 stato ACTIVE.

AWS CLI

Per creare una flotta di container con AWS CLI, apri una finestra della riga di comando e usa il `create-fleet` comando. Per ulteriori informazioni su questo comando, vedere [create-fleet](#) nella Guida ai AWS CLI comandi.

La `create-fleet` richiesta di esempio mostrata di seguito crea una nuova flotta di container con le seguenti caratteristiche:

- `ContainerGroupsConfiguration` Specifica una definizione di gruppo di contenitori a replica singola: `MegaFrogRaceServer.NA.v2` Tre copie del gruppo di repliche verranno distribuite su ciascuna istanza del parco istanze. Ogni istanza ha 30 porte di connessione disponibili per l'accesso ai processi sull'istanza.
- La flotta utilizza istanze `c5.large` On-Demand.
- Distribuisce gruppi di container nelle seguenti posizioni:

- us-west-2 (regione d'origine)
- ca-central-1 (posizione remota)
- Ogni gruppo di contenitori di repliche su un'istanza eseguirà contemporaneamente 5 processi del server di gioco, consentendo a ciascuna istanza di ospitare fino a 15 sessioni di gioco alla volta.
- In ogni gruppo di contenitori di repliche, Amazon GameLift consente l'attivazione di due nuove sessioni di gioco contemporaneamente. Inoltre, interrompe qualsiasi sessione di gioco di attivazione se non sono pronti a ospitare giocatori entro 300 secondi.
- La protezione è attivata su tutte le sessioni di gioco in hosting sulle istanze in questo parco istanze.
- I singoli giocatori possono creare tre nuove sessioni di gioco entro un intervallo di tempo di 15 minuti.

```
aws gamelift create-fleet \
  --name SampleFleet123 \
  --description "The sample test fleet" \
  --compute-type "CONTAINER" \
  --container-groups-configuration
  "ContainerGroupDefinitionNames=['MegaFrogRaceServer.NA.v2'],
  DesiredReplicaContainerGroupPerInstance=3,
  ConnectionPortRange={FromPort=1010,ToPort=1040}" \
  --ec2-instance-type c5.large \
  --region us-west-2 \
  --locations "Location=ca-central-1" \
  --fleet-type ON_DEMAND \
  --runtime-configuration "GameSessionActivationTimeoutSeconds=300,
  MaxConcurrentGameSessionActivations=2, ServerProcesses=[{LaunchPath=/local/game/
  MegaFrogRace/server.exe,ConcurrentExecutions=5}]" \
  --new-game-session-protection-policy "FullProtection" \
  --resource-creation-limit-policy "NewGameSessionsPerCreator=3,
  PolicyPeriodInMinutes=15" \
  --ec2-inbound-permissions
  "FromPort=1010,ToPort=1040,IpRange=0.0.0.0/0,Protocol=UDP" \
```

Se la richiesta di creazione della flotta ha esito positivo, Amazon GameLift restituisce una serie di attributi del parco veicoli che include le impostazioni di configurazione richieste e un nuovo ID della flotta. Amazon imposta GameLift quindi lo stato della flotta e lo stato della posizione

su Nuovo e avvia il processo di attivazione del parco veicoli. 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](#)

Gestisci le tue flotte di GameLift container Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

Se desideri ottenere informazioni sulla tua flotta di container o apportare modifiche, puoi utilizzare le seguenti azioni per gestire la tua flotta di container.

Visualizzazione delle risorse

Di seguito sono riportati alcuni modi per ottenere informazioni sulle risorse della flotta di container.

- [DescribeCompute](#)- Restituisce i dettagli su un contenitore registrato come calcolo.
- [DescribeContainerGroupDefinition](#)- Restituisce dettagli sulla definizione di un gruppo di contenitori. Questa risorsa descrive come sono configurati il gruppo e i relativi contenitori.
- [DescribeFleetAttributes](#)- Ottiene gli attributi della flotta, che includono l'intervallo delle porte di connessione e altri attributi.
- [DescribeFleetCapacity](#)- Ottiene un conteggio dei gruppi di container di replica nella flotta e dei relativi stati.
- [DescribeRuntimeConfiguration](#)- Descrive i processi del server eseguiti in ogni gruppo di contenitori di replica.
- [GetComputeAccess](#)- Fornisce l'accesso remoto a un'istanza che ospita il gruppo di contenitori.
- [GetComputeAuthToken](#)- Richiede un token di autenticazione da Amazon GameLift per una risorsa di calcolo in una flotta di container.
- [ListCompute](#)- Elenca i gruppi di contenitori registrati come calcoli.
- [ListContainerGroupDefinitions](#)- Elenca le definizioni dei gruppi di contenitori.
- [ListFleets](#)- Elenca le flotte che utilizzano uno specifico gruppo di container.

Aggiornare le risorse

Di seguito sono riportati alcuni modi in cui è possibile creare e modificare le risorse della flotta di container.

- [CreateContainerGroupDefinition](#)- Crea una definizione di gruppo di contenitori.
- [CreateFleet](#)- Crea una flotta di container quando ComputeType è impostato suCONTAINER.
- [RegisterCompute](#)- Registra i calcoli con una flotta di container.
- [UpdateFleetAttributes](#)- Aggiorna gli attributi mutabili di una flotta, come le opzioni di configurazione della flotta Anywhere.
- [UpdateFleetCapacity](#)- Aggiorna le impostazioni di capacità per una flotta EC2 gestita o una flotta di container.
- [UpdateRuntimeConfiguration](#)- Aggiorna la configurazione di runtime, che descrive quali processi server eseguire in ogni gruppo di contenitori di replica registrato come elaborazione.

Delete resources (Elimina risorse)

Di seguito sono riportati alcuni modi per rimuovere le risorse del parco container.

- [DeleteContainerGroupDefinition](#)- Elimina una definizione di gruppo di contenitori.
- [DeleteFleet](#)- Elimina una flotta.
- [DeregisterCompute](#)- Rimuove una risorsa di calcolo da una flotta di container.

Scalabilità delle flotte di GameLift container Amazon

Questa documentazione riguarda una funzionalità disponibile in anteprima pubblica. ed è soggetta a modifiche.

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, puoi scalare la capacità della flotta aggiungendo o rimuovendo istanze della flotta.

Quando crei una nuova flotta, Amazon GameLift imposta la capacità desiderata della flotta su un'istanza e distribuisce un'istanza nella regione di origine della flotta. Per una flotta con più sedi, Amazon GameLift distribuisce un'istanza nella regione di origine e in ciascuna località remota. Una volta raggiunto lo stato della flotta ACTIVE, puoi aumentare la capacità desiderata di scalabilità verso l'alto o ridurre la capacità desiderata di ridimensionamento.

Puoi utilizzare le funzionalità di GameLift scalabilità di Amazon per modificare la capacità manualmente o configurare il ridimensionamento automatico in base alla domanda dei giocatori:

- Imposta il ridimensionamento automatico con tracciamento del target. Per informazioni, consulta [Scalabilità automatica basata sugli obiettivi](#).
- Modifica manualmente la capacità della tua flotta. Per informazioni, consulta [Imposta manualmente la capacità per una GameLift flotta Amazon](#).

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 processes per replica container group] * [# of replica container groups per instance]
```

- Per i processi per gruppo di contenitori di repliche, chiama [DescribeRuntimeConfiguration](#) conta il numero di esecuzioni simultanee per i processi del server di gioco.
- Per i gruppi di contenitori di replica per istanza, chiama per [DescribeFleetAttributes](#) ottenere il valore. `DesiredReplicaContainerGroupPerInstance` Se questo valore non è impostato, usa il `MaxReplicaContainerGroupsPerInstance` valore.
- 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:

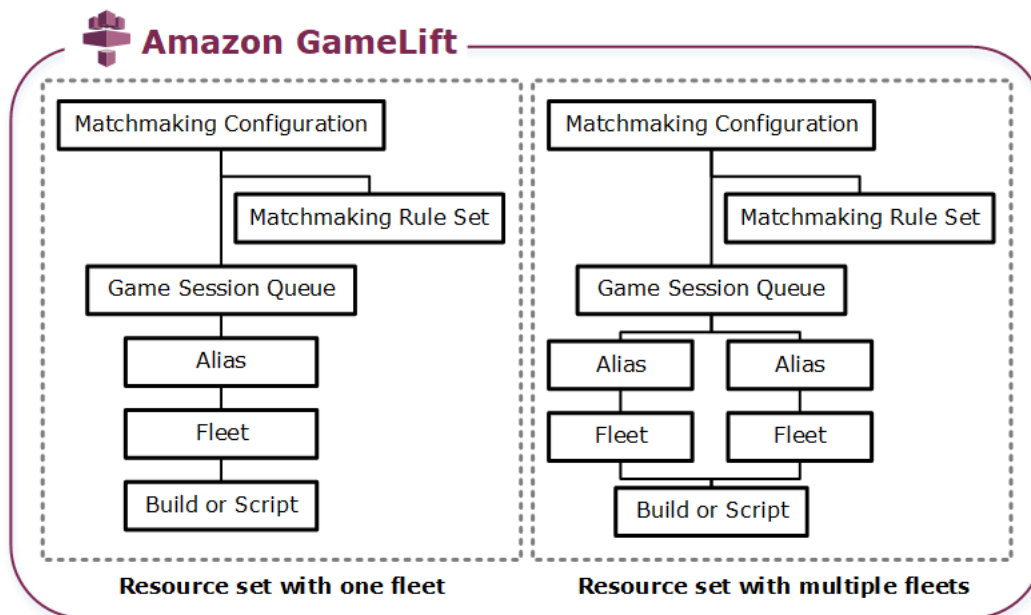
```
[Players per instance] = [# of game sessions per instance] * [# of player slots per game session]
```

Per ottenere l'attuale capacità totale di una flotta di container, chiama [DescribeFleetCapacity](#) o [DescribeFleetLocation Capacity](#) per ottenere il numero di gruppi di container di repliche 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 repliche.

Gestione delle risorse GameLift di hosting Amazon

Questa sezione fornisce informazioni dettagliate sulla configurazione delle risorse GameLift gestite da Amazon per gestire i server di gioco e ospitare sessioni di gioco per i giocatori. Devi configurare e distribuire le risorse, scalare la capacità per soddisfare la domanda dei giocatori e individuare le risorse disponibili per ospitare sessioni di gioco.

Il diagramma seguente illustra come gli oggetti GameLift delle risorse Amazon si relazionano tra loro. Usa una build o uno script per creare una flotta, assegnare un alias a una flotta e aggiungere flotte alla coda di una sessione di gioco utilizzando il relativo alias. Per i giochi che utilizzano il FlexMatch matchmaking, usa la coda delle sessioni di gioco e un set di regole di matchmaking per creare una configurazione di matchmaking.



Codice del server di gioco

- **Crea:** il tuo software per server di gioco personalizzato che funziona su Amazon GameLift e ospita sessioni di gioco per i tuoi giocatori. Una build di gioco rappresenta l'insieme di file che eseguono il tuo server di gioco su un particolare sistema operativo e che devi integrare con AmazonGameLift. Carica i file di creazione del gioco su Amazon GameLift nel Regione AWS luogo in cui intendi creare le flotte. Per ulteriori informazioni, consulta [Carica un server personalizzato su Amazon GameLift](#).
- **Script:** configurazione e logica di gioco personalizzate da utilizzare con server in tempo reale. Configura i server in tempo reale 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 [Carica uno script Realtime Servers su Amazon GameLift](#).

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 è possibile distribuire le flotte, vedere. [Sedi GameLift di hosting Amazon](#) Per informazioni sulla creazione di flotte, vedere [Configurazione delle GameLift flotte Amazon](#).

Alias

Un identificatore astratto per una flotta che puoi usare per cambiare la flotta a cui i tuoi giocatori sono connessi in qualsiasi momento. Per ulteriori informazioni, consulta [Aggiungi un alias a una flotta Amazon GameLift](#).

Coda delle sessioni di gioco

Un meccanismo di posizionamento delle sessioni di gioco che riceve 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. [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#)

Caricamento di build e script su Amazon GameLift

Prima di distribuire i tuoi server di gioco multiplayer per l'hosting con AmazonGameLift, devi caricare i file del server di gioco. Gli argomenti di questa sezione forniscono indicazioni sulla preparazione e il caricamento di file di build di server di gioco personalizzati o file di script del server Realtime Servers.

Argomenti

- [Carica un server personalizzato su Amazon GameLift](#)
- [Carica uno script Realtime Servers su Amazon GameLift](#)

Carica un server personalizzato su Amazon GameLift

Dopo aver integrato il tuo server di gioco con Amazon GameLift, carica i file di build su Amazon GameLift. Questo argomento spiega come impacchettare i file di build del gioco, creare uno script di installazione della build opzionale e quindi caricare i file utilizzando [AWS Command Line Interface\(AWS CLI\)](#) o un AWS SDK.

Argomenti

- [Creare un pacchetto dei file della build di gioco](#)
- [Crea una build Amazon GameLift](#)
- [Aggiorna i tuoi file di build](#)
- [Aggiungere uno script di installazione della build](#)

Creare un pacchetto dei file della build di gioco

Prima di caricare il server di gioco configurato su Amazon GameLift, impacchetta i file di build del gioco in una directory di build. Questa directory deve includere tutti i componenti necessari per l'esecuzione dei server di gioco e delle sessioni di gioco in hosting, fra cui quelli riportati di seguito:

- 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, consulta [Supporto allo sviluppo con Amazon GameLift](#).
- 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 Amazon GameLift Server SDK per C++ (incluse quelle create con il plug-in Unreal), includi la DLL OpenSSL per la stessa versione di OpenSSL con cui hai creato l'SDK del server. Consulta il file README dell'SDK del server per maggiori dettagli.

- Script di installazione (opzionale): un file di script per gestire le attività di installazione della build del gioco sui server di GameLift hosting Amazon. Posiziona questo file nella radice della directory di build. Amazon GameLift 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 in modo sicuro alle tue risorse su altri AWS servizi. Per informazioni su come eseguire questa operazione, consulta [Comunica 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. Ciò verifica che tu includa tutte le dipendenze richieste nel pacchetto e che lo script di installazione sia accurato.

Crea una build Amazon GameLift

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 crea una nuova risorsa di build con un ID di build univoco e altri metadati. La build inizia con lo stato Inizializzato. Dopo che Amazon ha GameLift acquisito i file del server di gioco, la build passa allo stato Ready.

Quando la build è pronta, puoi distribuirla su una nuova flotta Amazon GameLift . Per ulteriori informazioni, consulta [Crea una flotta GameLift gestita da Amazon](#) .Quando Amazon GameLift 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 e carica i file da una posizione specificata.

Inviare una richiesta di caricamento. In una finestra della riga di comando, inserisci 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 Amazon GameLift server SDK 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 dell'SDK del GameLift server Amazon con cui è integrato il tuo server di gioco. Se non fornisci un valore, Amazon GameLift utilizza il valore `4.0.2` predefinito. Se specifichi una versione SDK del server non corretta, la compilazione del server di gioco potrebbe

non riuscire durante la chiamata `InitSdk` per stabilire una connessione al GameLift servizio Amazon.

- `build-root`— Il percorso della directory dei tuoi 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 desideri creare la 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 fornisce lo stato di avanzamento del caricamento. Una volta completato correttamente il caricamento, Amazon GameLift 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 da lì. Quando crei la build, specifichi la posizione del bucket S3 e Amazon GameLift 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 compressi 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 una build Amazon. GameLift
2. Consenti ad Amazon di GameLift accedere ai tuoi file di build. Crea un ruolo IAM seguendo le istruzioni riportate in [Accedi 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 la GameLift console Amazon o il AWS CLI per creare un nuovo record di build. Devi avere l'PassRole autorizzazione, come descritto in [Esempi di autorizzazioni IAM per Amazon GameLift](#).

Console

1. Nella [GameLiftconsole Amazon](#), 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 per consentire ad Amazon di GameLift accedere al tuo bucket S3 e all'oggetto di compilazione.
4. (Facoltativo) In Tag, aggiungi tag alla build inserendo le coppie Key e Value.
5. Seleziona Create (Crea).

Amazon GameLift 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 Amazon GameLift server SDK 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 dell'SDK del GameLift server Amazon che hai usato per integrare il tuo server di gioco con Amazon. GameLift Se non fornisci un valore, Amazon GameLift utilizza il valore 4.0.2 predefinito.
- 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 i tuoi file di build

Puoi aggiornare i metadati per una risorsa di compilazione utilizzando la GameLift console Amazon o il [update-build](#) AWS CLI comando.

Dopo aver creato una GameLift build Amazon, non puoi aggiornare i file di build ad essa associati. Per ogni nuovo set di file, crea una nuova GameLift build Amazon. Utilizzando il [upload-build](#) comando, Amazon crea GameLift automaticamente un nuovo record di build per ogni richiesta. Se fornisci file di build utilizzando il [create-build](#) comando, carica un nuovo file.zip di build con un nome diverso su Amazon S3 e crea una build facendo riferimento al nuovo nome di file.

Per la distribuzione di build aggiornate, prova questi suggerimenti:

- Utilizza le code e scambia i parchi istanze in base alle esigenze. Quando configuri il tuo client di gioco con Amazon GameLift, specifica una coda anziché una flotta. Con le code, puoi aggiungere alla coda le nuove flotte con la nuova build e rimuovere le vecchie flotte. Per ulteriori informazioni, consulta [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#).

- Usa gli alias per trasferire i giocatori a una nuova build di gioco. Quando integri il tuo client di gioco con Amazon GameLift, specifica un alias di flotta anziché un ID della flotta. Per ulteriori informazioni, consulta [Aggiungi un alias a una flotta Amazon GameLift](#).
- Imposta aggiornamenti automatici delle build. Per esempi di script e informazioni su come incorporare le GameLift distribuzioni di Amazon nel tuo sistema di compilazione, consulta [Automating Deployments to Amazon](#) sul blog di Game Tech. GameLift AWS

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 installa la build e ricrea le directory dei file nel tuo 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 utilizzando `sudo`. 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 supporta linguaggi di interpretazione shell comuni come `bash`. Aggiungi uno shebang (ad esempio `#!/bin/bash`) all'inizio dello script di installazione. Per verificare il supporto per i tuoi comandi di shell preferiti, accedi in remoto a un'istanza Linux attiva e apri un prompt della shell. Per ulteriori informazioni, consulta [Connessione remota alle istanze GameLift della flotta Amazon](#).

- Lo script di installazione non può fare affidamento su una connessione peering VPC. Una connessione peering VPC è disponibile solo dopo l'installazione di GameLift da parte di Amazon delle istanze build on fleet.

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.

```
vc redistrib_x64.exe /install /quiet /norestart /log c:\game\vc redistrib_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'agente Amazon CloudWatch per raccogliere parametri personalizzati e a livello di sistema e gestire la rotazione dei log. Poiché Amazon GameLift funziona in un VPC di servizio, devi concedere ad Amazon GameLift le autorizzazioni per assumere un ruolo AWS Identity and Access Management (IAM) per tuo conto. Per consentire GameLift ad Amazon di assumere un ruolo, crea un ruolo che includa la politica AWS CloudWatchAgentAdminPolicy gestita e utilizzalo 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

```

Carica uno script Realtime Servers su Amazon GameLift

Quando sei pronto per distribuire server in tempo reale per il tuo gioco, carica i file di script del server Realtime completi su Amazon. GameLift Puoi farlo creando una risorsa di GameLift script Amazon e specificando la posizione dei tuoi 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 crei una nuova risorsa di script, Amazon GameLift le 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 dello script, Amazon GameLift distribuisce lo script con una nuova flotta di Realtime Servers. Amazon GameLift installa lo script del tuo server su ogni istanza del parco istanze, inserendo i file di `/local/game` script.

Per risolvere i problemi di attivazione della flotta relativi allo script del server, vedere. [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#)

File di script del pacchetto

Lo script del server può includere uno o più file combinati in un unico file.zip per il caricamento. Il file.zip deve contenere tutti i file necessari per l'esecuzione dello script.

Puoi archiviare i tuoi file di script compressi in una directory di file locale o in un bucket Amazon Simple Storage Service (Amazon S3).

Caricare file di script da una directory locale

Se i tuoi file di script sono archiviati localmente, puoi caricarli su Amazon GameLift da lì. Per creare la risorsa script, usa la GameLift console Amazon o il [AWS Command Line Interface\(AWS CLI\)](#).

Amazon GameLift console

Per creare una risorsa di script

1. Apri la [GameLiftconsole Amazon](#).
2. Nel riquadro di navigazione, scegli Hosting, Script.
3. Nella pagina Script, scegli Crea script.
4. Nella pagina Crea script, in Impostazioni dello script, procedi come segue:
 - a. In Nome, inserisci un nome di script.
 - b. (Facoltativo) In Versione, inserisci le informazioni sulla versione. Poiché puoi aggiornare il contenuto di uno script, i dati di versione possono essere utili per tenere traccia degli aggiornamenti.
 - c. Per Origine dello script, scegli Carica un file.zip.
 - d. Per i file di script, scegliete Scegli file, cercate il file.zip che contiene lo script, quindi scegliete quel file.

5. (Facoltativo) In Tag, aggiungi i tag allo script inserendo le coppie Chiave e Value.
6. Seleziona Create (Crea).

Amazon GameLift assegna un ID al nuovo script e carica il file.zip designato. Puoi visualizzare il nuovo script, incluso il suo stato, nella pagina Script.

AWS CLI

Usa il [create-script](#) AWS CLI comando per definire il nuovo script e caricare i file di script del server.

Per creare una risorsa di script

1. Posiziona 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 che Amazon GameLift elabori 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 tua richiesta, Amazon GameLift restituisce il nuovo oggetto script.

4. Per visualizzare il nuovo script, chiama [describe-script](#).

Caricare file di script da Amazon S3

Puoi archiviare i tuoi file di script in un bucket Amazon S3 e caricarli su Amazon GameLift da lì. Quando crei lo script, specifichi la posizione del bucket S3 e Amazon GameLift recupera 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 tuo server e caricalo in un bucket S3 in un bucket Account AWS che controlli. Prendi nota dell'URI dell'oggetto: ne hai bisogno quando crei uno script AmazonGameLift.

Note

Amazon GameLift non supporta il caricamento da bucket S3 con nomi che contengono un punto (.).

2. GameLiftConcedi ad Amazon l'accesso ai tuoi file di script. Per creare un ruolo AWS Identity and Access Management (IAM) che GameLift consenta ad Amazon di accedere al bucket S3 contenente lo script del server, segui le istruzioni in [Configura un ruolo di servizio IAM per Amazon GameLift](#). Dopo aver creato il nuovo ruolo, prendi nota del suo nome, che ti serve per creare uno script.
3. Creare uno script. Usa la GameLift console Amazon o il AWS CLI per creare un nuovo record di script. Per effettuare questa richiesta, è necessario disporre dell'PassRole autorizzazione IAM, come descritto in [Esempi di autorizzazioni IAM per Amazon GameLift](#).

Amazon GameLift console

1. Nella [GameLiftconsole Amazon](#), nel pannello di navigazione, scegli Hosting, Scripts.
2. Nella pagina Script, scegli Crea script.
3. Nella pagina Crea script, in Impostazioni dello script, procedi come segue:
 - a. In Nome, inserisci un nome di script.
 - b. (Facoltativo) In Versione, inserisci le informazioni sulla versione. Poiché puoi aggiornare il contenuto di uno script, i dati di versione possono essere utili per tenere traccia degli aggiornamenti.
 - c. Per l'origine dello script, scegli l'URI 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 di Amazon S3 e la versione dell'oggetto, scegli Browse S3, quindi cerca l'oggetto script.
4. (Facoltativo) In Tag, aggiungi i tag allo script inserendo le coppie Chiave e Value.
5. Seleziona Create (Crea).

Amazon GameLift assegna un ID al nuovo script e carica il file.zip designato. Puoi visualizzare il nuovo script, incluso il suo stato, nella pagina Script.

AWS CLI

Usa il [create-script](#) AWS CLI comando per definire il nuovo script e caricare i file di script del server.

1. Apri una finestra della riga di comando e passa a una directory in cui puoi usare il AWS CLI.
2. Immettete il create-script comando e i parametri seguenti. Il --storage-location parametro specifica la posizione del bucket Amazon S3 dei tuoi 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 restituisce il nuovo oggetto script.

3. Per visualizzare il nuovo script, chiama [describe-script](#).

Aggiornamento dei file di script

Puoi aggiornare i metadati per una risorsa di script utilizzando la GameLift console Amazon o il [update-script](#) AWS CLI comando.

Puoi anche aggiornare il contenuto dello script per una risorsa di script. Amazon GameLift distribuisce il contenuto degli script su tutte le istanze della flotta che utilizzano la risorsa di script aggiornata. Quando viene distribuito lo script aggiornato, le istanze lo utilizzano all'avvio di 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 dello script aggiornato, utilizza la GameLift console Amazon 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 verifica GameLift periodicamente la presenza di file di script aggiornati e li recupera direttamente dal bucket S3.

Configurazione delle GameLift flotte Amazon

Questa sezione fornisce informazioni dettagliate sulla progettazione, la costruzione e la manutenzione di flotte da utilizzare con Amazon GameLift. Puoi utilizzare le GameLift flotte Amazon per implementare server di gioco personalizzati e server in tempo reale.

Una flotta rappresenta le tue risorse di hosting come un set di istanze Amazon Elastic Compute Cloud (Amazon EC2) o hardware fisico. La posizione di una flotta determina dove vengono utilizzate le istanze o l'hardware per ospitare le sessioni di gioco per i tuoi giocatori. Le dimensioni di una flotta e il numero di sessioni di gioco e di giocatori che può supportare dipendono dal numero di istanze o dalla quantità di hardware che le fornisci. È possibile regolare le istanze virtuali manualmente o utilizzando il ridimensionamento automatico.

Molti giochi in produzione utilizzano più di una flotta. Puoi utilizzare più flotte, ad esempio, per far funzionare più di una versione del tuo server di gioco contemporaneamente, per fornire capacità di backup per Spot Fleets o per aumentare la ridondanza.

Per imparare a creare flotte progettate per le esigenze del tuo gioco, inizia con [Guida alla progettazione del GameLift parco veicoli Amazon](#). Dopo che la tua flotta è in funzione [Scalabilità](#)

[della capacità di GameLift hosting di Amazon](#), [vedi Aggiungi un alias a una flotta Amazon GameLift](#), e [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#).

Argomenti

- [Guida alla progettazione del GameLift parco veicoli Amazon](#)
- [Crea una nuova GameLift flotta Amazon](#)
- [Gestisci le tue GameLift flotte Amazon](#)
- [Aggiungi un alias a una flotta Amazon GameLift](#)
- [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#)
- [Connessione remota alle istanze GameLift della flotta Amazon](#)

Guida alla progettazione del GameLift parco veicoli Amazon

Questa guida alla progettazione illustra le migliori pratiche per creare una flotta di risorse di hosting da utilizzare con AmazonGameLift. Scegli una combinazione di risorse di hosting e scopri come configurarle in base al tuo gioco.

Argomenti

- [Scelta delle GameLift risorse di elaborazione Amazon](#)
- [Gestisci il modo in cui Amazon GameLift lancia i server di gioco](#)
- [Usa le istanze Spot con Amazon GameLift](#)

Scelta delle GameLift risorse di elaborazione Amazon

Per distribuire i tuoi server di gioco e ospitare sessioni di gioco per i tuoi giocatori, Amazon GameLift utilizza le risorse [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) chiamate istanze o il tuo hardware fisico. Quando configuri una nuova flotta utilizzando le istanze, decidi di che tipo di istanze ti servono e come eseguire i processi dei server di gioco su di esse. Quando una flotta EC2 gestita è attiva e pronta per ospitare sessioni di gioco, puoi aggiungere o rimuovere istanze secondo necessità per soddisfare la domanda dei giocatori.

Puoi distribuire i tuoi server di GameLift gioco Amazon su una combinazione di due tipi di elaborazione:

- **EC2 gestito:** le flotte EC2 gestite utilizzano istanze Amazon EC2 per ospitare i server di gioco. Amazon GameLift gestisce le istanze e rimuove l'onere della gestione hardware e software derivante dall'hosting dei tuoi giochi.
- **Amazon GameLift Anywhere:** GameLift Anywhere le flotte Amazon utilizzano la tua infrastruttura esistente per ospitare server di gioco mentre Amazon GameLift gestisce il matchmaking e le code.

Quando scegli le risorse di elaborazione per la tua flotta, considera i seguenti fattori:

- [Hardware disponibile](#)
- [Ubicazione della flotta](#)
- [Istanze On-Demand e istanze Spot](#)
- [Sistemi operativi](#)
- [Tipi di istanza](#)
- [Quote del servizio](#)

Hardware disponibile

Considerate l'infrastruttura esistente nella vostra implementazione. Mentre migri i giochi su Amazon GameLift, puoi continuare a utilizzare la tua infrastruttura. Con Amazon GameLift Anywhere, puoi utilizzare la tua infrastruttura insieme alle istanze EC2 GameLift gestite da Amazon. Puoi anche utilizzare la tua infrastruttura esistente per ospitare giochi più vicini ai tuoi giocatori rispetto a quanto consentito dalle GameLift sedi Amazon supportate. Per ulteriori informazioni sulla configurazione delle GameLift Anywhere flotte Amazon, consulta [Crea una GameLift Anywhere flotta Amazon](#).

Ubicazione della flotta

Considera le aree geografiche in cui intendi distribuire 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 [Sedi GameLift di hosting Amazon](#)

Per le GameLift Anywhere flotte Amazon, sei tu a determinare la posizione dell'hardware fisico. Per ulteriori informazioni sulle sedi personalizzate, consulta [Amazon GameLift Anywhere](#).

Istanze On-Demand e istanze Spot

Le istanze On-Demand di Amazon EC2 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 utilizza le code e l'algoritmo FleetIQ per determinare AWS che interromperà un'istanza Spot, e mette l'istanza in uno stato di riciclaggio. Quindi, quando non ci sono sessioni di gioco attive sull'istanza, Amazon GameLift prova a sostituirla.

Per ulteriori informazioni su come utilizzare le istanze Spot, consulta [Usa le istanze Spot con Amazon GameLift](#).

Sistemi operativi

GameLift Le istanze Amazon supportano build di server di gioco eseguite su Microsoft Windows o Amazon Linux. Quando carichi una build di gioco su Amazon GameLift, specifica il sistema operativo del gioco. Quando crei una flotta Amazon EC2 per distribuire la build del gioco, Amazon configura GameLift automaticamente le istanze con il sistema operativo della build. Per ulteriori informazioni sui sistemi operativi per server di gioco supportati, consulta. [Supporto allo sviluppo con Amazon GameLift](#)

Quando utilizzi una GameLift Anywhere flotta Amazon, puoi utilizzare qualsiasi sistema operativo supportato dal tuo hardware. GameLift AnywhereLe flotte Amazon richiedono che tu distribuisca la build del gioco sull'hardware mentre utilizzi Amazon GameLift per gestire le tue risorse in un unico posto.

Tipi di istanza

Il tipo di istanza di una flotta Amazon EC2 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 GameLift server Amazon basato 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 istanza, consulta la sezione Tipi di [istanze di Amazon EC2](#).
- Per informazioni sull'esecuzione di più processi per istanza, consulta [Gestisci il modo in cui Amazon GameLift lancia i server di gioco](#).

Quote del servizio

Per visualizzare le quote di servizio predefinite per Amazon GameLift e le quote correnti per il tuo Account AWS, procedi come segue:

- Per informazioni generali sulle quote di servizio per Amazon GameLift, consulta [GameLiftEndpoints e quote Amazon](#) nel. Riferimenti generali di AWS
- Per un elenco dei tipi di istanze disponibili per sede per il tuo account, apri la pagina delle [quote di servizio](#) della GameLift console Amazon. Questa pagina mostra anche l'uso 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 nella console [Amazon GameLift](#). Il GameLift team di Amazon utilizza il questionario di lancio per determinare le quote e i limiti corretti per il gioco.

Gestisci il modo in cui Amazon GameLift lancia i server di gioco

Puoi configurare la configurazione di runtime di una flotta EC2 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 utilizza la configurazione di runtime di un parco veicoli 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 server di gioco eseguibile. 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 server definiti nella configurazione di runtime. Con più processi, Amazon GameLift scaglionava l'avvio di ogni processo. I processi server hanno una durata limitata. Al termine, Amazon GameLift lancia nuovi processi per mantenere il numero e il tipo di processi 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 della flotta per implementare le modifiche. Ecco come Amazon GameLift adotta le modifiche alla configurazione del runtime:

1. L'istanza invia una richiesta ad Amazon GameLift per la versione più recente della configurazione di runtime.
2. L'istanza confronta i suoi 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 server, i processi server attivi di questo tipo continuano a essere eseguiti 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 essere eseguiti fino alla fine. L'istanza non sostituisce questi processi server in eccesso.
- Se la configurazione di runtime aggiornata aggiunge un nuovo tipo di processo server o aumenta i processi concorrenti per un tipo esistente, l'istanza avvia nuovi processi server, fino al GameLift massimo di Amazon. In questo caso, l'istanza avvia nuovi processi server quando terminano i 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 su un parco veicoli, quindi carica la build su Amazon. GameLift Tutti i server di gioco di una build devono funzionare sulla stessa piattaforma e utilizzare Amazon GameLift Server SDK.
- 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 possa riciclare i processi in modo efficiente. Ad 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 tuo server di gioco per gestire le richieste di terminazione da Amazon. GameLift
- Assicurati che Amazon si GameLift spenga e riavvii i processi del server non sani. Segnala lo stato di salute ad Amazon GameLift implementando la funzione di `OnHealthCheck()` callback nel codice del tuo server di gioco. Amazon GameLift interrompe automaticamente i processi del server segnalati non integri per tre report consecutivi. Se non esegui

l'implementazione `OnHealthCheck()`, Amazon GameLift presume che un processo del 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 limita ogni istanza a un [numero massimo di processi simultanei](#). La somma di tutti i processi simultanei per le configurazioni dei processi server di una flotta non può superare questa quota.
- Per mantenere livelli di prestazioni accettabili, il tipo di istanza Amazon EC2 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 non esegue più processi simultanei del numero totale configurato. Ciò significa che la transizione dalla precedente configurazione di runtime alla nuova configurazione potrebbe avvenire gradualmente.

Usa le istanze Spot con Amazon GameLift

Quando configuri la tua flotta EC2 GameLift gestita da Amazon, puoi utilizzare istanze Spot, istanze On-Demand o una combinazione. Scopri di più su come Amazon GameLift utilizza le istanze Spot in [Istanze On-Demand e istanze Spot](#). Per utilizzare le flotte spot, l'integrazione del gioco richiede le modifiche elencate in questa pagina.

Lo stai usando FlexMatch per il matchmaking? Puoi aggiungere parchi istanze Spot alle tue code di sessione di gioco esistenti per posizionamenti di abbinamento.

1. Progetta la coda delle sessioni di gioco per le istanze Spot.

Gestire il posizionamento delle sessioni di gioco con una coda è la migliore pratica ed è necessaria quando si utilizzano le istanze Spot. Per progettare la coda, considera quanto segue:

- Luoghi: per ottenere la migliore esperienza di gioco, scegli località geograficamente vicine ai tuoi giocatori.
- Tipi di istanze: considera i requisiti hardware dei tuoi server di gioco e la disponibilità delle istanze nelle località che hai scelto.

Per provare una coda che ottimizzi la disponibilità e la resilienza di Spot, consulta. [Tutorial: configura una coda di sessioni di gioco per le istanze Spot](#)

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 flotta GameLift gestita da Amazon](#) 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 di 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 tuo 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 trasmette la notifica a tutti i processi server interessati richiamando la funzione di callback di Amazon GameLift Server SDK. `onProcessTerminate()` Implementa questo callback per terminare la sessione di gioco o spostare la sessione di gioco e i giocatori su una nuova istanza. Consultare [Rispondere a una notifica di arresto del 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.

Visualizza i GameLift parametri di Amazon nella GameLift console Amazon o con Amazon CloudWatch per verificare le prestazioni. Per ulteriori informazioni sui GameLift parametri di Amazon, consulta [Monitora Amazon GameLift 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.

Crea una nuova GameLift flotta Amazon

Crea una nuova flotta e distribuisci la tua build di server di gioco personalizzata o server in tempo reale per l'hosting. Puoi distribuire qualsiasi build di gioco o risorsa di script che carichi su AmazonGameLift.

Argomenti

- [Come funziona la creazione di GameLift flotte Amazon](#)
- [Crea una flotta GameLift gestita da Amazon](#)
- [Crea una GameLift Anywhere flotta Amazon](#)

Come funziona la creazione di GameLift flotte Amazon

Quando crei un nuovo parco veicoli, Amazon GameLift avvia un flusso di lavoro che crea un parco veicoli con un'istanza Amazon Elastic Compute Cloud (Amazon EC2) in ogni sede del parco veicoli. Man mano che Amazon GameLift completa ogni fase del flusso di lavoro, la flotta emette eventi e Amazon GameLift aggiorna lo stato della flotta. Puoi tenere traccia di tutti gli eventi utilizzando la

GameLift console Amazon o richiamando l'operazione Amazon GameLift API [DescribeFleetEvents](#). Puoi anche monitorare lo stato delle singole sedi utilizzando [DescribeFleetLocationAttributes](#).

Flusso di lavoro per la creazione di flotte EC2:

- Amazon GameLift crea una risorsa della flotta nella regione di origine della flotta e in ogni postazione remota definita nella flotta.
- Amazon GameLift imposta la capacità desiderata su un'istanza.
- Amazon GameLift imposta lo stato della flotta e della sede su Nuovo.
- Amazon GameLift inizia a scrivere eventi nel registro eventi della flotta.
- Amazon GameLift alloca le risorse informatiche richieste per una nuova istanza in ogni sede del parco veicoli.
- Amazon GameLift scarica i file del server di gioco in ogni istanza e imposta lo stato della flotta su Download.
- Amazon GameLift convalida i file del server di gioco scaricati su ogni istanza per verificare che non si siano verificati errori durante il download. Amazon GameLift imposta lo stato della flotta su Convalida.
- Amazon GameLift crea il server di gioco su ogni istanza e imposta lo stato della flotta su Edificio.
- Amazon GameLift inizia a lanciare i processi del server su ogni istanza, seguendo le istruzioni nella configurazione di runtime della flotta. Se hai configurato il parco macchine per eseguire più processi server simultanei per istanza, Amazon GameLift scaglionava l'avvio del processo di alcuni secondi. Man mano che ogni processo viene messo online, segnala la disponibilità ad AmazonGameLift. Amazon GameLift imposta lo stato della flotta su Attivazione.
- Amazon GameLift imposta gli stati della flotta e gli stati della posizione su Attivo man mano che i processi del server segnalano la disponibilità.

Amazon GameLift Anywhere fleet creation

- Amazon GameLift crea una risorsa per la flotta. Per la regione di origine della flotta e per ogni sede personalizzata definita nella flotta, Amazon GameLift imposta lo stato della flotta e della posizione su Nuovo.
- Amazon GameLift inizia a scrivere eventi nel registro eventi della flotta.
- Dopo che un processo del server di una flotta notifica ad Amazon GameLift che è pronto, Amazon GameLift imposta lo stato della flotta e lo stato della posizione su Attivo. Man mano che i processi

dei server in altre sedi del parco veicoli segnalano la disponibilità, Amazon GameLift imposta lo stato di ogni sede del parco veicoli su Attivo.

Per assistenza nella risoluzione dei problemi relativi alla creazione di flotte, consulta [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#).

Crea una flotta GameLift gestita da Amazon

Usa la [GameLift console Amazon](#) o AWS Command Line Interface (AWS CLI) per creare una flotta gestita.

Dopo aver creato una nuova flotta EC2 gestita, lo stato della flotta passa attraverso diverse fasi, man mano che Amazon GameLift distribuisce la flotta e installa e avvia i server di gioco. Una volta raggiunto lo stato, la flotta è pronta per ospitare sessioni di gioco. ACTIVE Per assistenza in caso di problemi con la creazione del parco istanze, consulta [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#).

Console

Per creare una flotta EC2 gestita

1. Nella [GameLift console Amazon](#), nel pannello di navigazione, scegli Fleets.
2. Nella pagina Fleets (Parchi istanze) scegliere Create (Crea).
3. Scegli Managed EC2.
4. Nella pagina dei dettagli della flotta, procedi come segue:
 - a. Per Nome, inserisci il nome del parco veicoli. Ti consigliamo di includere il tipo di flotta (Spot o On-demand) nei nomi della flotta. In questo modo è molto più semplice identificare i tipi di flotta quando si visualizza un elenco di flotte.
 - b. Per Descrizione, fornisci una breve descrizione della flotta.
 - c. Per il tipo binario, seleziona Build o Script per definire il tipo di server di gioco che Amazon GameLift distribuisce a questa flotta.
 - d. Seleziona uno script o una build dall'elenco a discesa degli script o delle build caricati.
5. (Facoltativo) In Dettagli aggiuntivi per quanto segue:
 - a. Ad esempio, specifica un ruolo IAM che autorizzi le applicazioni della build del gioco ad accedere ad altre AWS risorse del tuo account. Per ulteriori informazioni, consulta

[Comunica con altre AWS risorse delle tue flotte](#). Per creare una flotta con un ruolo di istanza, il tuo account deve disporre dell'`PassRole` autorizzazione IAM. Per ulteriori informazioni, consulta [Esempi di autorizzazioni IAM per Amazon GameLift](#).


Se desideri autorizzare applicazioni che non sono eseguibili dal server, come un CloudWatch agente, abilita l'opzione delle credenziali condivise.

Non puoi aggiornare queste impostazioni dopo la creazione del parco veicoli.

- b. Per la generazione della certificazione, scegli di fare in modo che Amazon GameLift generi un certificato TLS per la flotta. Puoi utilizzare un certificato TLS del parco istanze per fare in modo che il client di gioco autentichi un server di gioco durante la connessione e crittografi tutte le comunicazioni client/server. Per ogni istanza in una flotta abilitata per TLS, Amazon crea GameLift anche una nuova voce DNS con il certificato. Utilizza queste risorse per configurare l'autenticazione e la crittografia per il gioco.
- c. Per Metric group, inserisci il nome di un gruppo di metriche della flotta nuovo o esistente. Puoi aggregare le metriche per più flotte aggiungendole allo stesso gruppo di metriche.

Non puoi aggiornare il gruppo di metriche dopo la creazione della flotta.

6. Seleziona Successivo.
7. Nella pagina Seleziona sedi, seleziona una o più postazioni remote aggiuntive in cui distribuire le istanze. La regione principale viene selezionata automaticamente in base alla regione da cui si accede alla console. Se selezioni sedi aggiuntive, anche le istanze del parco istanze vengono distribuite 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 Gestire in. Regioni AWS](#)
[Riferimenti generali di AWS](#)

8. Seleziona Successivo.
9. Nella pagina Definisci i dettagli dell'istanza, scegli
 - a. Istanze su richiesta o Spot per questo parco istanze. Per ulteriori informazioni sui tipi di parco veicoli, consulta [Istanze On-Demand e istanze Spot](#)
 - b. Dal menu Filter architecture scegli x64 o Arm.

 Note

Le istanze Graviton Arm richiedono un GameLift server Amazon 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).

[Per informazioni sulle architetture Arm di Amazon EC2, consulta Processore AWSGraviton e tipi di istanze Amazon EC2.](#)

[Per informazioni sui tipi di istanza supportati da Amazon GameLift, consulta i EC2InstanceType valori sotto CreateFleet \(\) request parameters.](#)

10. Seleziona un tipo di istanza Amazon EC2 dall'elenco. Per ulteriori informazioni sulla scelta di un tipo di istanza, consulta [Tipi di istanza](#). Dopo aver creato il parco istanze, non puoi modificare il tipo di istanza.
11. Seleziona Successivo.
12. Nella pagina Configura runtime, in Configurazione di runtime, procedi come segue:
 - a. Per Launch path, inserisci il percorso dell'eseguibile del gioco nella build o nello script. Sulle istanze Windows, i server di gioco vengono compilati 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, oMyRealtimeLaunchScript.js.**

- b. (Facoltativo) Per i parametri di avvio, inserisci le informazioni da passare all'eseguibile del gioco come set di parametri della riga di comando. Esempio: **+sv_port 33435 +start_lobby.**
- c. Per i processi simultanei, scegli il numero di processi server da eseguire contemporaneamente su ciascuna istanza del parco istanze. Verifica i GameLift [limiti](#) di Amazon sul numero di processi server simultanei.

Al totale dei processi simultanei, si applicano dei limiti sui processi simultanei del server per istanza per tutte le configurazioni. Se configuri la flotta in modo da superare il limite, la flotta non può attivarsi.

13. In Attivazione della sessione di gioco, fornisci i limiti per l'attivazione di nuove sessioni di gioco sulle istanze di questa flotta:
 - a. Per il numero massimo di sessioni di gioco simultanee, inserisci il numero massimo di sessioni di gioco su un'istanza che si attivano contemporaneamente. Questo limite è utile quando l'avvio di più nuove sessioni di gioco può influire sulle prestazioni delle altre sessioni di gioco in esecuzione sull'istanza.
 - b. Per Nuovo timeout di attivazione, inserisci il tempo di attesa per l'attivazione di una sessione. Se lo stato della sessione di gioco non passa ACTIVE allo stato precedente al timeout, Amazon GameLift interrompe l'attivazione della sessione di gioco.
14. (Facoltativo) Nelle impostazioni della porta EC2, procedi come segue:
 - a. Scegli Aggiungi impostazione porta per definire le autorizzazioni di accesso per il traffico in entrata che si connette al processo server distribuito sulla flotta.
 - b. Per Tipo, scegli TCP personalizzato o UDP personalizzato.
 - c. Per Intervallo di porte, inserisci un intervallo di numeri di porta che consentono le connessioni in entrata. Un intervallo di porte deve utilizzare il formato `nnnnn[-nnnnn]`, con valori compresi tra 1026 e 60000. Esempio: **1500** o **1500-20000**.
 - d. Per Intervallo di indirizzi IP, immettere un intervallo di indirizzi IP. Utilizzare la notazione CIDR. Esempio: **0.0.0.0/0** (questo esempio consente l'accesso a chiunque tenti di connettersi).
15. (Facoltativo) In Impostazioni delle risorse della sessione di gioco, procedi come segue:

- a. Per la politica di protezione di Game Scaling, attiva o disattiva la protezione di scalabilità. Amazon GameLift non interromperà l'istanza con protezione durante un evento di ridimensionamento se sta ospitando una sessione di gioco attiva.
 - b. Per Limite di creazione di risorse, inserisci un numero massimo di sessioni di gioco che un giocatore può creare durante il periodo di validità della polizza.
16. Seleziona Successivo.
 17. (Facoltativo) Aggiungi tag alla build inserendo le coppie Chiave e Valore. Scegli Avanti per continuare con la revisione della creazione della flotta.
 18. Seleziona Create (Crea). Amazon GameLift assegna un ID alla nuova flotta e avvia il processo di attivazione della flotta. È possibile monitorare lo stato del nuovo parco istanze nella pagina Fleets (Parchi istanze).

Puoi aggiornare i metadati e la configurazione della flotta in qualsiasi momento, indipendentemente dallo stato della flotta. Per ulteriori informazioni, consulta [Gestisci le tue GameLift flotte Amazon](#). Puoi aggiornare la capacità della flotta dopo che la flotta ha raggiunto lo stato ATTIVO. Per ulteriori informazioni, consulta [Scalabilità della capacità di GameLift hosting di Amazon](#). Puoi anche aggiungere o rimuovere postazioni remote.

AWS CLI

Per creare una flotta con AWS CLI, apri una finestra della riga di comando e usa il `create-fleet` comando. Per ulteriori informazioni sul comando `create-fleet`, consulta [create-fleet](#) in the Guida di riferimento ai comandi della AWS CLI.

La richiesta `create-fleet` di esempio riportata di seguito consente di creare un nuovo parco istanze con le seguenti caratteristiche:

- La flotta utilizza le istanze On-Demand `c5.large` con il sistema operativo appropriato per la build di gioco selezionata.
- Implementa la build del server di gioco specificata, che deve essere in stato Pronto, nelle seguenti posizioni:
 - `us-west-2` (regione d'origine)
 - `sa-east-1` (posizione remota)
- La generazione di certificati TLS è abilitata.
- Ogni istanza nel parco istanze eseguirà simultaneamente dieci processi identici del server di gioco e potrà ospitare fino a dieci sessioni di gioco in contemporanea.

- In ogni caso, Amazon GameLift consente l'attivazione di due nuove sessioni di gioco contemporaneamente. Inoltre, interrompe qualsiasi sessione di gioco di attivazione se non sono pronti a ospitare giocatori entro 300 secondi.
- La protezione è attivata su tutte le sessioni di gioco in hosting sulle istanze in questo parco istanze.
- I singoli giocatori possono creare tre nuove sessioni di gioco entro un intervallo di tempo di 15 minuti.
- Ogni sessione di gioco ospitata su questa flotta ha un punto di connessione che rientra negli intervalli di indirizzi IP e porte specificati.
- Amazon GameLift aggiunge i parametri per questa flotta al gruppo di `EMEAfleets` metriche, che (in questo esempio) combina i parametri per tutte le flotte nelle regioni EMEA.

```
aws gamelift create-fleet \
  --name SampleFleet123 \
  --description "The sample test fleet" \
  --ec2-instance-type c5.large \
  --region us-west-2 \
  --locations "Location=sa-east-1" \
  --fleet-type ON_DEMAND \
  --build-id build-92f061ed-27c9-4a02-b1f4-6f85b2385620 \
  --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=33435,IpRange=0.0.0.0/0,Protocol=UDP"
"FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP" \
  --metric-groups "EMEAfleets"
```

Se la richiesta di creazione della flotta ha esito positivo, Amazon GameLift restituisce un set di attributi del parco veicoli che include le impostazioni di configurazione richieste e un nuovo ID della flotta. Amazon avvia GameLift quindi il processo di attivazione della flotta e imposta lo stato della flotta e lo stato della sede 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](#)

Crea una GameLift Anywhere flotta Amazon

Usa Amazon GameLift per integrare l'hardware del tuo ambiente nel tuo hosting di GameLift giochi Amazon. Amazon GameLift Anywhere registra il tuo hardware presso Amazon GameLift in una Anywhere flotta. Puoi integrare Anywhere e gestire le flotte EC2 nelle code dei matchmaker e delle sessioni di gioco per gestire il matchmaking e il posizionamento delle partite.

Per ulteriori informazioni su come testare i server di gioco con Amazon GameLift Anywhere, consulta [Testa la tua integrazione utilizzando le GameLift Anywhere flotte Amazon](#).

Per iniziare, utilizza la [Supporto allo sviluppo con Amazon GameLift](#) versione 5 o successiva e consulta i seguenti concetti per l'utilizzo di una GameLift Anywhere flotta Amazon.

Posizioni personalizzate

GameLift AnywhereLe flotte Amazon utilizzano posizioni personalizzate per rappresentare le ubicazioni fisiche della tua infrastruttura.

Registrazione del dispositivo

Per consentire a una GameLift Anywhere flotta Amazon di comunicare con le tue risorse di elaborazione, devi prima registrare il dispositivo. Puoi completare la registrazione del dispositivo da Amazon GameLift AWS SDK utilizzando l'[RegisterCompute](#) operazione. Questa operazione utilizza l'indirizzo IP del dispositivo per associarlo a una posizione della flotta e comunicare con Amazon GameLift.

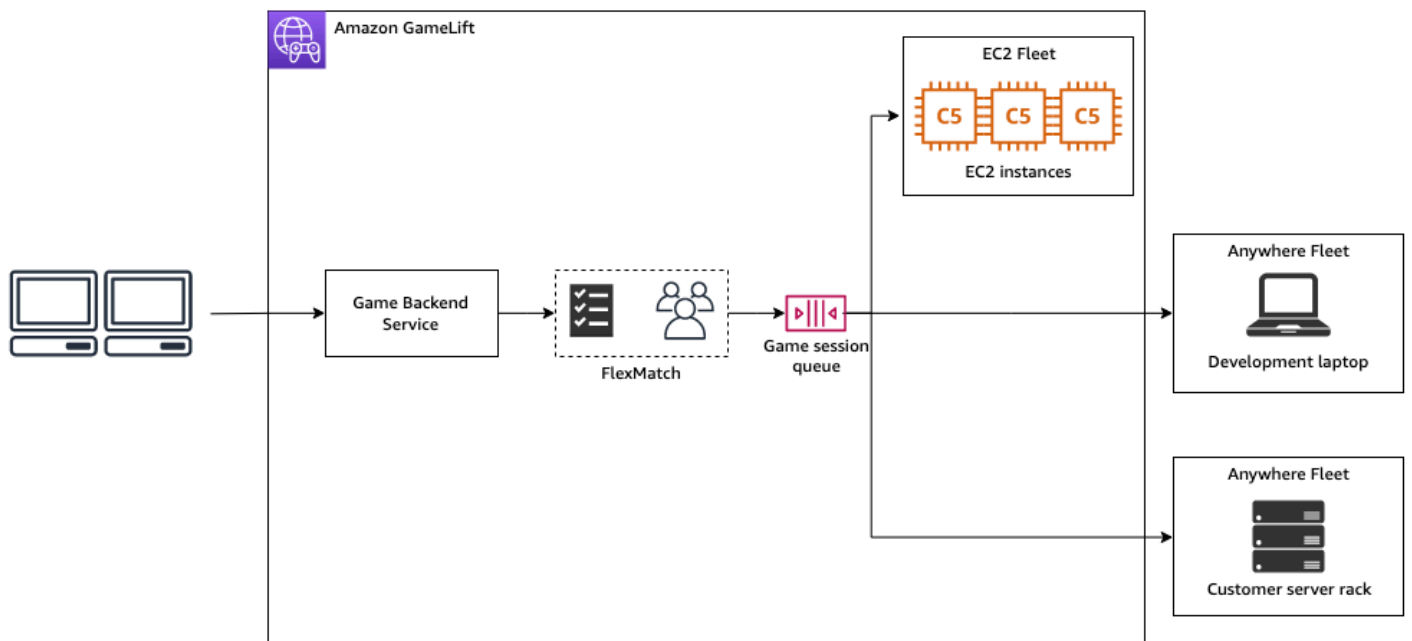
Token di autenticazione

Quando inizi un server di gioco sul tuo computer, l'Amazon GameLift Server SDK utilizza un token di autenticazione per autenticare il tuo server di gioco su Amazon. GameLift Puoi riutilizzare lo stesso token di autenticazione per tutti i server di gioco sullo stesso computer, fino alla scadenza del token di autenticazione. Per recuperare il token di autenticazione, chiama il comando (). [get-compute-auth-token](#) AWS Command Line Interface AWS CLI Passa il token a ciascun server di gioco secondo necessità.

Sessioni di gioco

Ogni sessione di gioco su un computer utilizza lo stesso token di autenticazione creato durante la registrazione del calcolo in una postazione della flotta.

Il diagramma seguente mostra una coda di sessioni di gioco che utilizza FlexMatch matchmaking e più flotte. Le flotte includono una flotta EC2 con istanze C5, una flotta con un laptop di sviluppo e una Anywhere flotta con un server rack ospitato dal cliente. Anywhere



Argomenti

- [Crea una posizione personalizzata](#)
- [Creazione di un parco istanze](#)
- [Registra il tuo computer](#)
- [Esegui un processo sul server](#)
- [Creare sessioni di gioco](#)
- [Esegui la migrazione a EC2 gestito](#)

Crea una posizione personalizzata

Per iniziare a ospitare giochi sulle tue risorse di elaborazione, crea una posizione personalizzata che descriva dove risiede il tuo computer.

Console

Per creare una posizione personalizzata

1. Apri la [GameLift console Amazon](#).
2. Nel pannello di navigazione, sotto Hosting, scegli Sedi.
3. Nella pagina Locations (Posizioni) scegliere Create location (Crea posizione).
4. Nella finestra di dialogo Crea posizione, procedi come segue:

- a. Immettete il nome di una località. Indica la posizione del tuo hardware che Amazon GameLift utilizza per eseguire i tuoi giochi nelle Anywhere flotte. Amazon GameLift aggiunge il nome della tua sede personalizzata con custom-.
- b. (Facoltativo) Aggiungi tag come coppie chiave-valore alla tua posizione personalizzata. Scegli Aggiungi nuovo tag per ogni tag che desideri aggiungere.
- c. Scegli Crea.

AWS CLI

Crea una posizione personalizzata utilizzando il [create-location](#) comando. location-name Indica la posizione dell'hardware GameLift utilizzato da Amazon per eseguire i giochi nelle Anywhere flotte. Quando crei una località personalizzata, il nome della località deve iniziare con custom-.

```
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"  
  }  
}
```

Creazione di un parco istanze

Usa la [GameLift console Amazon](#) o il AWS CLI per creare una Anywhere flotta.

Dopo aver creato una nuova Anywhere flotta, lo stato della flotta passa da NEW a ACTIVE. Quando raggiunge lo ACTIVE stato, la flotta è pronta per ospitare sessioni di gioco. Per assistenza in caso di problemi con la creazione del parco istanze, consulta [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#).

Console

Per creare una Anywhere flotta

1. Apri la [GameLift console Amazon](#).
2. Nel pannello di navigazione, sotto Hosting, scegli Fleets.
3. Nella pagina Fleets (Parchi istanze) scegliere Create (Crea).
4. Nella fase Tipo di calcolo, scegli Anywhere, quindi scegli Avanti.
5. Nella fase Dettagli della flotta, definisci i dettagli, quindi scegli Avanti.
6. Nella fase Posizioni personalizzate, seleziona la posizione personalizzata che hai creato, quindi scegli Avanti. Amazon seleziona GameLift automaticamente la casa Regione AWS come regione in cui stai creando la flotta. Puoi utilizzare la regione d'origine per accedere e utilizzare le tue risorse.
7. Completa i passaggi rimanenti per la creazione della flotta, quindi scegli Invia per creare la tua Anywhere flotta.

AWS CLI

Crea una Anywhere flotta usando il [create-fleet](#) comando. Includi la tua posizione personalizzata in `locations`. Amazon GameLift crea la flotta nella tua regione di residenza e nelle località personalizzate che fornisci. Nell'esempio seguente, sostituisci *FleetName* e *custom-location-1* con le tue informazioni. La variabile `custom-location-1` è il nome della posizione creata nel [Crea una posizione personalizzata](#) passaggio.

```
aws gamelift create-fleet \  
--name FleetName \  
--compute-type ANYWHERE \  
--locations "Location=custom-location-1"
```

Output di esempio

```
{  
  "FleetAttributes": {  
    "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",  
    "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445",  
    "Name": "HardwareAnywhere",  
    "CreationTime": "2023-02-23T17:57:42.293000+00:00",
```

```

    "Status": "ACTIVE",
    "MetricGroups": [
      "default"
    ],
    "CertificateConfiguration": {
      "CertificateType": "DISABLED"
    },
    "ComputeType": "ANYWHERE"
  }
}

```

Registra il tuo computer

Per registrare la tua risorsa di calcolo nella flotta che hai creato, usa il [register-compute](#) comando. *fleet-id* Sostituiscilo con l'ARN della flotta *fleet-id* restituito nel passaggio precedente o trovato nella pagina dei dettagli del tuo parco veicoli nella console. Sostituisci e *ip-address* con l'indirizzo IP della tua risorsa di elaborazione. *compute-name*

Note

Ti consigliamo di chiamare entrambi i `get-compute-auth-token` comandi `register-compute` e da uno script o da un gestore di processi diverso dal server di gioco.

```

aws gamelift register-compute \
  --compute-name HardwareAnywhere \
  --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445 \
  --ip-address 10.1.2.3 \
  --location custom-location-1

```

Output di esempio

```

{
  "Compute": {
    "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",
    "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445",
    "ComputeName": "HardwareAnywhere",
  }
}

```



```
    "ComputeArn": "arn:aws:gamelift:us-east-1:111122223333: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-east-1.api.amazongamelift.com"  
  }  
}
```

Esegui un processo sul server

1. Ottieni il token di autenticazione per la tua risorsa di calcolo dalla flotta che hai creato.

Il tuo server di gioco utilizza il token di autenticazione per l'autenticazione con Amazon GameLift. Ogni token di autenticazione ha una scadenza. Per continuare a utilizzare la risorsa di calcolo per ospitare il server di gioco, recupera un nuovo token di autenticazione prima della scadenza.

Note

Amazon GameLift consiglia di chiamare entrambi i `get-compute-auth-token` comandi `register-compute` e da uno script o un gestore di processi separato dal server di gioco.

Nell'esempio seguente, sostituisci il `fleet-id` con l'ARN o l'ID flotta del parco veicoli creato nei passaggi precedenti. Sostituisci il `compute-name` con il nome del calcolo creato utilizzando il `register-compute` comando in un passaggio precedente.

```
aws gamelift get-compute-auth-token \  
  --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445 \  
  --compute-name HardwareAnywhere
```

Output di esempio:

```
{  
  "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",  
  "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445",
```

```

    "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"
}

```

2. Esegui un'istanza del file eseguibile del tuo server di gioco.

Per far funzionare il server di gioco, inizializza il server di gioco chiamandolo `InitSDK()` e passandogli i parametri del server. Per ulteriori informazioni, consulta [ServerParameters](#).

Ingresso SDK del server:

```

//Define the server parameters
ServerParameters serverParameters = new ServerParameters(
    websocketUrl=wss://us-east-1.api.amazongamelift.com,
    processId=PID1234,
    hostId=HardwareAnywhere,
    fleetId=arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-
cebb4da2-52a8-4c27-9b85-587f945c6445,
    authToken=0c728041-3e84-4aaa-b927-a0fb202684c0);

//InitSDK establishes a connection with GameLift's websocket server for
communication.
var initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);

```

3. Quando il processo server è pronto per ospitare una sessione di gioco, chiama `AmazonProcessReady()` dal server di gioco GameLift. Per ulteriori informazioni sui parametri di processo, consulta [ProcessParameters](#)

```

// Set parameters and call ProcessReady
var processParams = new ProcessParameters(
    this.OnStartGameSession,
    this.OnProcessTerminate,
    this.OnHealthCheck,
    this.OnUpdateGameSession,
    port=1024,
    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);
```

Creare sessioni di gioco

1. Aggiungi la logica al tuo server di gioco in modo che il processo del server risponda al `onStartGameSession()` messaggio con `activateGameSession()`. Questa operazione non ha parametri, ma invia una conferma ad Amazon GameLift che il tuo server ha ricevuto e accettato il messaggio di creazione della sessione di gioco.

```
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();  
}
```

2. Dal servizio di backend del client di gioco, avvia la sessione di gioco utilizzando il comando [start-matchmakingstart-game-session-placement](#), o. [create-game-session](#)

```
aws gamelift create-game-session \  
  --fleet-id arn:aws:gamelift:us-east-1:682428703967:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445 \  
  --name GameSession1 \  
  --maximum-player-session-count 2 \  
  --location custom-location-1
```

Output di esempio:

```
GameSession {  
  FleetId = arn:aws:gamelift:us-east-1:682428703967:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445,  
  GameSessionId = 4444-4444,  
  Name = GameSession1,  
  Location = custom-location-1,  
  IPAddress = 10.2.3.4,  
  Port = 1024,  
  ...
```

```
}
```

Amazon GameLift invia un `onStartGameSession()` messaggio al processo del tuo server registrato. Il messaggio contiene l'oggetto `GameSession` del passaggio precedente con le proprietà del gioco, i dati delle sessioni di gioco, i dati del matchmaker e altro ancora sulla sessione di gioco.

3. Quando la sessione di gioco è completa, termina il processo del server di gioco.

Ingresso SDK del server:

```
var processEndingOutcome = GameLiftServerAPI.ProcessEnding();
if (processReadyOutcome.Success)
    Environment.Exit(0);
// otherwise, exit with error code
Environment.Exit(errorCode);
```

4. Avvia un altro processo del server di gioco `ProcessReady(processParams)` chiamando.

Esegui la migrazione a EC2 gestito

Dopo aver sviluppato il server di gioco e aver completato la preparazione per la produzione, puoi affidare la GameLift gestione dell'hardware ad Amazon. Per migrare a una flotta EC2 gestita, carica la tua build su Amazon GameLift e crea una flotta EC2 gestita. Per ulteriori informazioni sul caricamento della build e sulla configurazione di una flotta, consulta e. [Carica un server personalizzato su Amazon GameLift](#) [Crea una flotta GameLift gestita da Amazon](#)

Gestisci le tue GameLift flotte Amazon

Usa la GameLift console Amazon o l'interfaccia a AWS riga di comando per aggiornare le impostazioni del parco veicoli, modificare le sedi remote o eliminare una flotta.

Aggiornare la configurazione di una flotta

Puoi aggiornare gli attributi mutabili della flotta, le impostazioni delle porte e le configurazioni di runtime utilizzando la GameLift console Amazon o l'AWSinterfaccia a riga di comando. Per modificare i limiti di scala, vedere [Ridimensiona automaticamente la capacità della flotta con Amazon GameLift](#).

Amazon GameLift console

1. Nella [GameLiftconsole Amazon](#), nel riquadro di navigazione, scegli Flotte.

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 metrici, che Amazon CloudWatch utilizza per tenere traccia delle GameLift metriche Amazon aggregate per più flotte.
 - Aggiorna le impostazioni del limite di creazione delle risorse.
 - Attiva o disattiva la protezione delle sessioni di gioco.
 - Configurazione di runtime: è possibile modificare una delle seguenti impostazioni delle configurazioni di runtime e aggiungere o rimuovere le 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 tuoi 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.
 - Impostazioni delle porte EC2: 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 un parco veicoli:

- [update-fleet-attributes](#)
- [update-fleet-port-settings](#)
- [update-runtime-configuration](#)

Aggiorna le posizioni della flotta

Puoi aggiungere o rimuovere le sedi remote di una flotta utilizzando la GameLift console Amazon o l'interfaccia a AWS riga di comando. Non puoi cambiare la regione di origine di una flotta.

Amazon GameLift console

1. Nella [GameLiftconsole Amazon](#), nel riquadro di navigazione, scegli Flotte.
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, seleziona la scheda Sedi per visualizzare le sedi della flotta.
4. Per aggiungere nuove sedi remote, scegli Aggiungi e seleziona le posizioni in cui desideri distribuire le istanze. Questo elenco non include i casi in cui il tipo di istanza del parco veicoli non è disponibile.
5. Con le nuove sedi selezionate, scegli Aggiungi. Amazon GameLift aggiunge le nuove sedi all'elenco, con lo stato impostato suNEW. Amazon inizia GameLift quindi a fornire un'istanza in ogni posizione aggiunta e a prepararla per ospitare sessioni di gioco.
6. Per rimuovere le sedi remote esistenti dalla flotta, utilizza le caselle di controllo per selezionare una o più sedi elencate.
7. Con una o più flotte selezionate, scegli Rimuovi. Le posizioni rimosse rimangono nell'elenco, con lo stato impostato suDELETING. Amazon avvia GameLift quindi il processo di interruzione dell'attività nella posizione rimossa. Se ci sono istanze attive che ospitano sessioni di gioco, Amazon GameLift utilizza la procedura di terminazione del server di gioco per terminare correttamente le sessioni di gioco, chiudere i server di gioco e chiudere le istanze.

AWS CLI

Utilizza i seguenti comandi AWS CLI per aggiornare le sedi della flotta:

- [create-fleet-locations](#)
- [delete-fleet-locations](#)

Eliminare una flotta

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 elimina la connessione peering VPC durante l'eliminazione della flotta.

Puoi utilizzare la GameLift console Amazon o lo strumento AWS CLI per eliminare un parco veicoli.

Amazon GameLift console

1. Nella [GameLiftconsole Amazon](#), nel riquadro di navigazione, scegli Flotte.
2. Scegli la flotta che desideri eliminare. Puoi eliminare le flotte solo nel ACTIVE nostro ERROR stato.
3. Scegliere Elimina.
4. Nella finestra di dialogo Elimina flotta, conferma l'eliminazione **delete** inserendo.
5. Scegliere Elimina.

AWS CLI

Utilizza il seguente comando AWS CLI per eliminare un parco veicoli:

- [delete-fleet](#)

Aggiungi un alias a una flotta Amazon GameLift

Un GameLift alias Amazon viene utilizzato per astrarre la designazione di una flotta. Le designazioni delle flotte indicano ad Amazon GameLift dove cercare le risorse disponibili quando si creano nuove sessioni di gioco per i giocatori. Usa gli alias anziché gli ID specifici della flotta per trasferire

senza problemi il traffico dei giocatori da una flotta all'altra modificando la posizione di destinazione dell'alias.

Per gli alias sono disponibili due tipi di strategie di routing:

- Semplice: indirizza il traffico dei giocatori verso un ID flotta specifico. 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.

Le flotte hanno una durata limitata e ci sono diversi motivi per cambiarle durante la durata di una partita. Non puoi aggiornare la configurazione del server di gioco di una flotta o modificare determinati attributi delle risorse di calcolo su una flotta esistente. Invece, crea nuove flotte con le modifiche e poi trasferisci i giocatori alle nuove flotte. Con gli alias, la sostituzione dei parchi istanze ha un impatto minimo sul gioco ed è completamente trasparente per i giocatori.

Gli alias sono utili nei giochi che non utilizzano code. La sostituzione di parchi istanze in una coda si effettua semplicemente creando un nuovo parco, aggiungendolo alla coda e rimuovendo il parco vecchio, senza che i giocatori si accorgano di nulla. Al contrario, i client di gioco che non utilizzano code devono specificare quale flotta utilizzare per comunicare con il servizio Amazon GameLift. Senza alias, un fleet switch richiede l'aggiornamento del codice di gioco e, possibilmente, la distribuzione di client di gioco aggiornati ai giocatori.

Quando si aggiorna il fleet-id a cui fa riferimento un alias, c'è un periodo di transizione di massimo 2 minuti durante il quale le sessioni di gioco con l'alias possono finire sulla vecchia flotta.

Crea un nuovo alias

[Puoi creare un alias utilizzando la GameLift console Amazon, come descritto qui, o con il comando AWS CLI create-alias.](#)

1. Nella [GameLift console Amazon](#), nel pannello di navigazione, scegli Alias.
2. Nella pagina Alias scegli Crea alias. Ti consigliamo di includere il tipo di flotta nei tuoi nomi alias. In questo modo è molto più semplice identificare il tipo di flotta quando si visualizza un elenco di alias.
3. Nella pagina Crea alias, in Dettagli alias, procedi come segue:
 - a. Per Nome, inserisci un nome alias.

- b. Per la descrizione, inserire una breve descrizione per l'identificazione.
 - c. Scegliete il tipo di routing semplice o terminale.
4. (Facoltativo) In Tag, aggiungi tag all'alias inserendo le coppie Chiave e Valore.
5. Scegli Crea.

Modifica un alias

[Puoi modificare un alias utilizzando la GameLift console Amazon o con il comando AWS CLI `update-alias`.](#)

1. Nella [GameLift console Amazon](#), nel pannello di navigazione, scegli Alias.
2. Nella pagina Alias, scegli l'alias che desideri modificare.
3. Nella pagina degli alias scegli Modifica.
4. Nella pagina Edit alias (Modifica alias), è possibile modificare gli elementi seguenti:
 - Nome alias: nome descrittivo per il tuo alias.
 - Descrizione: breve descrizione del tuo alias.
 - Tipo: strategia di routing per il traffico dei giocatori. Selezionare Simple (Semplice) per modificare il parco istanze associato oppure Terminal (Terminale) per modificare il messaggio di interruzione.
5. Scegli Save changes (Salva modifiche).

Esegui il debug dei problemi relativi alla GameLift flotta Amazon

Questo argomento fornisce indicazioni sui problemi di configurazione della flotta per una soluzione di hosting GameLift gestito Amazon. Per ulteriori informazioni sulla risoluzione dei problemi, puoi accedere da remoto a un'istanza del parco istanze quando è attivo. Consultare [Connessione remota alle istanze GameLift della flotta Amazon](#).

Problemi di creazione del parco istanze

Quando viene creata una flotta, il GameLift servizio Amazon avvia un flusso di lavoro che distribuisce una nuova istanza in ciascuna delle sedi della flotta e la prepara a far funzionare i tuoi server di gioco. Per una descrizione dettagliata, vedere [Come funziona la creazione di GameLift flotte Amazon](#). Una flotta non può ospitare sessioni di gioco e giocatori finché non raggiunge lo stato Attivo. Questa sezione illustra i problemi più comuni che impediscono alle flotte di diventare attive.

Downloading and validating (In fase di download e convalida)

Durante questa fase, la creazione della flotta potrebbe non riuscire se ci sono problemi con i file di build estratti, lo script di installazione non viene eseguito o se i file eseguibili designati nella configurazione di runtime non sono inclusi nei file di build. Amazon GameLift fornisce registri relativi a ciascuno di questi problemi.

Se nei log non sono segnalati errori, è probabile che il problema sia dovuto a un errore del servizio interno. In questo caso, prova a creare nuovamente il parco istanze. Se il problema persiste, prova a ricaricare la build del gioco, in caso i file fossero danneggiati. Puoi anche contattare l'GameLiftassistenza Amazon o pubblicare una domanda sul forum.

In fase di creazione

I problemi che causano errori durante la fase di creazione di modelli sono quasi certamente dovuti ai file della build del gioco e/o allo script di installazione. Verifica che i file di build del gioco, così come caricati su AmazonGameLift, possano essere installati su una macchina con il sistema operativo appropriato. Utilizza un'installazione del sistema operativo pulita e non un ambiente di sviluppo esistente.

Activating (In fase di attivazione)

I problemi più comuni di creazione del parco istanze si verificano durante la fase Activating (In fase di attivazione). Durante questa fase, vengono testati diversi elementi, tra cui la fattibilità del server di gioco, le impostazioni di configurazione del runtime e la capacità del server di gioco di interagire con il GameLift servizio Amazon utilizzando il Server SDK. Tra i problemi comuni riscontrabili durante l'attivazione del parco istanze:

Errore di avvio dei processi del server.

Verifica innanzitutto di avere impostato correttamente il percorso di avvio e i parametri di avvio opzionali nella configurazione di runtime del parco istanze. Puoi visualizzare la configurazione di runtime corrente della flotta utilizzando la pagina dei dettagli del parco veicoli, [Informazioni](#) () o chiamando il AWS CLI comando [describe-runtime-configuration](#). Se la configurazione di runtime sembra corretta, controlla la presenza di eventuali problemi nei file della build del gioco e/o negli script di installazione.

I processi del server si avviano, ma il parco istanze non si attiva.

Se i processi del server vengono avviati ed eseguiti correttamente, ma la flotta non passa allo stato Attivo, è probabile che il processo del server non comunichi ad Amazon GameLift

che è pronto per ospitare sessioni di gioco. Controlla che il server di gioco stia richiamando correttamente l'operazione dell'API server `ProcessReady()` (vedi [Inizializza il processo del server](#)).

La 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). Puoi monitorare l'esito positivo o negativo della richiesta di peering [describe-vpc-peering-connections](#) chiamando. Assicurati di verificare che esista un'autorizzazione 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 GameLift console Amazon (modifica le impostazioni di allocazione della capacità del parco veicoli) o chiamando il AWS CLI comando [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 appositamente connessioni peering VPC (vedere. [Peering VPC per Amazon GameLift](#)) 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. Devi prima ottenere un'autorizzazione chiamando l'API del GameLift servizio Amazon [CreateVpcPeeringAuthorization\(\)](#) o utilizzando il comando AWS CLI. `create-vpc-peering-authorization` Una volta ottenuta l'autorizzazione, è possibile eliminare il parco istanze.

Problemi relativi alla flotta di server in tempo reale

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 degli script non vengono rilevati dai server Realtime della flotta.
- 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.

Questo è quasi certamente il risultato della mancata chiamata dello `processEnding` 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 per avviarne una nuova e le nuove sessioni di gioco possono essere avviate solo quando vengono attivati nuovi server Realtime. Inoltre, gli aggiornamenti allo script Realtime non influiscono sulle sessioni di gioco già in esecuzione, ma solo su quelle.

Per evitare che si verifichi una situazione simile, gli script devono fornire un meccanismo di attivazione per la chiamata `processEnding`. Come illustrato negli [Esempio di script Realtime Servers](#), 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 rientri in questo scenario, ci sono un paio di soluzioni alternative per sbloccare i tuoi server in tempo reale. Il trucco consiste nell'attivare il riavvio dei processi del server in tempo reale, o delle istanze della flotta sottostanti. In questo caso, chiude GameLift automaticamente le sessioni di gioco per te. Una volta liberati, i server Realtime possono avviare nuove sessioni di gioco utilizzando l'ultima versione 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. Questo cancellerà tutte le sessioni di gioco esistenti e ti consentirà di ricominciare da capo con 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 alle istanze GameLift della flotta Amazon](#).

Se scegli di non includere la modalità di chiamata `processEnding` nel tuo script in tempo reale, ci sono un paio di situazioni difficili che potrebbero verificarsi anche quando la flotta si attiva e vengono avviate le sessioni di gioco. 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 riceve alcun aggiornamento degli script.

Connessione remota alle istanze GameLift della flotta Amazon

Puoi connetterti a qualsiasi istanza delle tue flotte Amazon GameLift gestite da EC2 attive. I motivi più comuni per accedere 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 alle istanze in flotte attive. Le flotte non attive, quelle in fase di attivazione o in stato di errore, potrebbero essere accessibili per un breve periodo di tempo. Per assistenza con i problemi di attivazione della flotta, consulta. [Esegui il debug dei problemi relativi alla GameLift flotta Amazon](#)

- 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 e ospita la sessione di gioco. Potrebbe spegnersi 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'AWSSDK, come documentato nel riferimento all'API del [GameLift servizio Amazon](#).

Raccogli dati delle istanze

Raccogli le seguenti informazioni:

- L'ID dell'istanza a cui desideri connetterti. È possibile utilizzare l'ID dell'istanza o l'ARN.
- La versione dell'SDK GameLift del server Amazon utilizzata sull'istanza. L'SDK del server è integrato con la build del gioco in esecuzione sull'istanza.

Per recuperare i dati dell'istanza

I passaggi seguenti presuppongono che tu disponga di un Fleet ID EC2 gestito per l'istanza a cui desideri connetterti.

1. Ottieni il nome del computer.

Chiama [list-compute](#) per la flotta EC2 gestita per ottenere un elenco di tutti i computer attivi nel parco. Per un parco veicoli con sede singola, specifica l'ID della flotta o l'ARN. Per una flotta con più sedi, specifica l'ID della flotta o l'ARN e una posizione. Per un parco EC2 gestito, i calcoli sono istanze EC2 e la proprietà restituita è l'ID dell'istanza. ComputeName Per esempio:

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",
      "Type": "c4.large"
    }
  ]
}
```

2. Trova la versione SDK del server.

La versione SDK del server è un attributo di una risorsa di compilazione.

- Chiama [describe-fleet-attributes](#) con un ID flotta per ottenere l'ID di build e l'ARN della flotta.
- 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_2",
  "ServerSdkVersion": "5.1.1",
  . . .
}
```

Connect a un'istanza (server SDK 5)

Se l'istanza a cui desideri connetterti esegue una build di gioco con server SDK versione 5.x, usa le seguenti istruzioni per connetterti all'istanza utilizzando Amazon EC2 Systems Manager (SSM). Puoi accedere alle istanze remote in esecuzione su Windows o Linux.

1. Richiedere le credenziali di accesso all'istanza. Quando disponi di un nome di calcolo e di un ID di flotta per l'istanza a cui desideri connetterti, chiama. [get-compute-access](#) In caso di successo, Amazon GameLift 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. Facoltativamente, puoi 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, consulta [Avvio di una sessione \(AWSCLI\)](#) nella Guida per l'utente di Amazon EC2 Systems Manager. Utilizza le credenziali acquisite nella fase 1. Per esempio:

```
aws ssm start-session \
--target i-11111111a222b333c \
--region us-west-2
```

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-access](#) per richiedere le credenziali di accesso. In caso di successo, Amazon GameLift 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 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-----
nEXAMPLEKEYKCAQEAY7WZhaDsR1W3mR1QtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
\nvBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfwIw5/akH7i05dSrvC7dQkW2duV5QuUdE0QW
\nZ/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449i1x9X1F
\nG50TCFe0zf18dqqCP6GzbPaIjiU19xX/az0R9V+tpU0zEL+wmXnZt3/nHPQ5xvD20JH67km6SuPW
\noPzev/D8V+x4+bHthfSjR9Y7DvQFjfbVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrrqu
\n/uler7vgIn5m71N5LKw4hJLAIW6tUT/fzvtCHK0SkbQCQXuriHmQ2MqyJX/0kn2NfjLV/
ufGxbL1\nmb5qwMGUnEpJaZD6QSSs3kICLwUYUiGfc0uiSbmJoap/
GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2\nbahyWyJNfjLe4M86yd2YK3V2CmK+X/
B0sShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/
tJWSD9\n81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA30zdXzMqexXVJ1TLZVEH0E7bh1Y9d801ozR
\noQs/FiZNAx2iijCwyv01pjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfq1
+1Ip1\nYkriL0DbLXlvRAH+yHPRit2hH0jtUNZh4Axv+cpq09qbUI3+43eEy24B7G/Uh
+GTfbjsXs0xQx/x\np9otyVwc7hsQ5TA5PZb
+mvkJ50BEKzet9XCkKw0NBVELGhnEPe7cCgYEA06Vgov6YH1eHui9kHuws
```

```

\nayav0e1c5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URipzWV0Eihvm+xTtmaZ1Sp//1kq75XDwnU
\nWA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
\nngYBjb0+0Zk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
\noMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiWiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
\nArq6Wv/G16zQuAE9zK9vwwKBgF+09VI/1wJBirsDGz9whVwFFPrTkJNvJZzYt69qezx1sjgFKshy
\nWBhd4xHZtmCqpBP1AymEjr/T01bxyARmXmNIOWIANNXMGB4KGSy11mzSVAoQ+fqR+cJ3d0dyP11j
\njjb0Ed/NY8fr1NDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLda
\nNWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvTW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
\nVRkAKKKYeGjKpUfVTirW0YFjXkfcir/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=\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 nelle GameLift flotte Amazon tramite qualsiasi porta autorizzata nella configurazione del parco veicoli. 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 che hai caricato su Amazon GameLift. Includono uno o più file eseguibili, risorse e dipendenze dei 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.
- Risorse GameLift di hosting Amazon. La directory principale `Whitewater` contiene i file utilizzati dal GameLift servizio Amazon 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'AWSSDK [UpdateRuntimeConfiguration](#) o 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/

Scalabilità della capacità di GameLift hosting di Amazon

La capacità di hosting, misurata in istanze, rappresenta il numero di sessioni di gioco che Amazon GameLift può 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 aumentare 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. Durante la visualizzazione o il ridimensionamento della capacità, le informazioni vengono elencate per posizione, inclusa la regione di origine della flotta ed eventuali sedi remote aggiuntive.

Puoi impostare manualmente il numero di istanze da mantenere oppure puoi impostare il ridimensionamento automatico per regolare dinamicamente la capacità al variare della domanda dei giocatori. Ti consigliamo di iniziare attivando l'opzione di ridimensionamento automatico basato sul target. L'obiettivo dell'auto scaling basato sugli obiettivi è mantenere risorse di hosting sufficienti per accogliere i giocatori attuali, oltre a un piccolo extra per gestire picchi imprevisti della domanda dei giocatori. Per la maggior parte dei giochi, il ridimensionamento automatico basato sugli obiettivi offre una soluzione di ridimensionamento altamente efficace.

Gli argomenti di questa sezione offrono informazioni dettagliate sulle attività seguenti:

- [Impostare i limiti minimo e massimo per il dimensionamento della capacità](#)

- [Impostare manualmente i livelli di capacità](#)
- [Usa il ridimensionamento automatico basato sugli obiettivi](#)
- [Gestisci il ridimensionamento automatico basato su regole \(funzionalità avanzata\)](#)
- [Disattiva temporaneamente il ridimensionamento automatico](#)

Puoi eseguire la maggior parte delle attività di scalabilità della flotta utilizzando la GameLift console Amazon. Puoi anche utilizzare un AWS SDK o il AWS Command Line Interface (AWS CLI) con l'[API del GameLift servizio Amazon](#).

Per gestire la capacità della flotta nella console

1. Apri la [GameLiftconsole Amazon](#).
2. Nel pannello di navigazione, scegli Hosting, Flotte.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Scegliete la scheda Ridimensionamento. In questa scheda puoi:
 - Visualizza le metriche storiche di scalabilità per l'intera flotta.
 - Visualizza e aggiorna le impostazioni di capacità per ogni sede della flotta, inclusi i limiti di scalabilità e le impostazioni attuali della capacità.
 - Aggiorna il ridimensionamento automatico basato sugli obiettivi, visualizza le politiche di scalabilità automatica basate su regole applicate all'intero parco veicoli e sospendi l'attività di ridimensionamento automatico per ogni sede.

Argomenti

- [Imposta i limiti GameLift di capacità di Amazon](#)
- [Imposta manualmente la capacità per una GameLift flotta Amazon](#)
- [Ridimensiona automaticamente la capacità della flotta con Amazon GameLift](#)

Imposta i limiti GameLift di capacità di Amazon

Per scalare la capacità di hosting di una GameLift flotta Amazon, manualmente o automaticamente, considera i limiti di scalabilità della sede. Tutte le sedi della flotta hanno un limite minimo e massimo che definisce l'intervallo consentito per la capacità della sede. Per impostazione predefinita, i limiti

sulle sedi del parco veicoli prevedono un minimo di 0 istanze e un massimo di 1 istanza. Prima di poter scalare la posizione di una flotta, modifica i limiti.

Se utilizzi la scalabilità automatica, il limite massimo consente GameLift ad Amazon di ampliare la flotta per soddisfare la domanda dei giocatori, ma evita costi di hosting eccessivi, ad esempio durante un attacco DDOS. Configura un [CloudWatchallarme Amazon](#) per avvisarti quando la capacità si avvicina al limite massimo, in modo da poter valutare la situazione e modificarla manualmente in base alle 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 limiti di capacità per le sedi di una flotta nella [GameLiftconsole Amazon](#) o utilizzando AWS Command Line Interface (AWS CLI).

Per impostare i limiti di capacità

Console

1. Apri la [GameLiftconsole Amazon](#).
2. Nel pannello di navigazione, scegli Hosting, Flotte.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Nella scheda Ridimensionamento, in Capacità di ridimensionamento, seleziona una sede della flotta, quindi scegli Modifica.
5. Nella finestra di dialogo Modifica capacità di ridimensionamento, impostate i conteggi delle istanze per Dimensione minima, Istanze desiderate e Dimensione massima.
6. Scegli Conferma.

AWS CLI

1. Controlla le impostazioni attuali della capacità. In una finestra della riga di comando, usa il [describe-fleet-location-capacity](#) comando con l'ID della flotta e l'ubicazione di cui desideri modificare la capacità. Questo comando restituisce un [FleetCapacity](#) oggetto che include le impostazioni di capacità correnti della posizione. Determina se i nuovi limiti di istanza possono soddisfare l'impostazione corrente delle istanze desiderate.

```
aws gamelift describe-fleet-location-capacity \  
  --fleet-id <fleet identifier> \  
  --location <location> \  
  --output <output type> \  
  --query <query>
```

```
--location <location name>
```

2. Update limit settings (Aggiornare le impostazioni dei limiti). In una finestra della riga di comando, utilizzare il [update-fleet-capacity](#) comando 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 tua richiesta va a buon fine, Amazon GameLift restituisce l'ID del parco veicoli. Se il nuovo `min-size` valore `max-size` o il valore sono in conflitto con l'`desired-instances` impostazione corrente, Amazon GameLift restituisce un errore.

Imposta manualmente la capacità per una GameLift flotta Amazon

Quando crei una nuova flotta, Amazon imposta GameLift automaticamente le istanze desiderate su un'istanza in ogni sede del parco veicoli. Quindi, Amazon GameLift distribuisce una nuova istanza in ogni sede. Per modificare la capacità della flotta, puoi aggiungere una politica di scalabilità automatica basata sugli obiettivi 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 necessario il ridimensionamento automatico 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 scalabilità automatica basata sugli obiettivi. Se disponi di una politica di ridimensionamento automatico

basata sugli obiettivi, ripristina immediatamente la capacità desiderata in base alle proprie regole di ridimensionamento.

Puoi impostare manualmente la capacità nella GameLift console Amazon o utilizzando AWS Command Line Interface (AWS CLI). Lo stato della flotta deve essere attivo.

Sospendi il ridimensionamento automatico

Puoi sospendere tutte le attività di ridimensionamento automatico per ogni sede della flotta. Con la sospensione del ridimensionamento automatico, il numero desiderato di istanze nella sede del parco veicoli rimane invariato, a meno che non venga modificato manualmente. Quando sospendi la scalabilità automatica per una sede, ciò influisce sulle politiche attuali della flotta e su tutte le politiche che potresti definire in futuro.

Per impostare manualmente la capacità del parco istanze

Console

1. Apri la [GameLiftconsole Amazon](#).
2. Nel pannello di navigazione, scegli Hosting, Flotte.
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 ridimensionamento automatico sospese, seleziona ogni posizione per cui desideri sospendere il ridimensionamento automatico, quindi scegli Sospendi.
5. In Capacità di ridimensionamento, seleziona una posizione che desideri impostare manualmente, quindi scegli Modifica.
6. Nella finestra di dialogo Modifica capacità di ridimensionamento, impostate il valore preferito per le istanze desiderate, quindi scegliete Conferma. Questo indica ad Amazon GameLift il numero di istanze da mantenere attive, pronte per ospitare sessioni di gioco.

Amazon GameLift risponde alle modifiche implementando istanze aggiuntive o chiudendo quelle non necessarie. Man mano che Amazon GameLift completa questo processo, il numero di istanze attive nella posizione cambia per corrispondere al valore aggiornato delle istanze desiderato. Questo processo può richiedere alcuni istanti.

AWS CLI

1. Controlla le impostazioni attuali della capacità. In una finestra della riga di comando, usa il [describe-fleet-location-capacity](#) comando con l'ID della flotta e l'ubicazione di cui desideri modificare la capacità. Questo comando restituisce un [FleetCapacity](#) oggetto che include le impostazioni di capacità correnti della posizione. Determina se i limiti delle istanze possono soddisfare la nuova impostazione delle istanze desiderate.

```
aws gamelift describe-fleet-location-capacity \  
  --fleet-id <fleet identifier> \  
  --location <location name>
```

2. Aggiornare la capacità desiderata. Usa il [update-fleet-capacity](#) comando con l'ID della flotta, la posizione e un nuovo valore per le istanze desiderate. Se questo valore non rientra nell'intervallo limite corrente, è possibile modificare 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 tua richiesta va a buon fine, Amazon GameLift restituisce l'ID del parco veicoli. Se l'impostazione delle nuove istanze desiderate non rientra nei limiti minimo e massimo, Amazon GameLift restituisce un errore.

Ridimensiona automaticamente la capacità della flotta con Amazon GameLift

Usa la scalabilità automatica in Amazon GameLift per scalare dinamicamente la capacità della tua flotta in risposta all'attività del server di gioco. Man mano che i giocatori arrivano e iniziano le sessioni di gioco, l'auto scaling può aggiungere altre istanze; man mano che la domanda dei giocatori diminuisce, l'auto scaling 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, devi creare politiche di scalabilità che indicano ad Amazon GameLift quando aumentare o diminuire la scalabilità. Esistono due tipi di politiche di scalabilità: basate sugli obiettivi e basate su regole. L'approccio basato sugli obiettivi, il tracciamento degli obiettivi, è una soluzione completa. La consigliamo come opzione più semplice ed efficace. Le politiche di scalabilità basate su regole richiedono di definire ogni aspetto del processo decisionale di scalabilità automatica, utile per risolvere problemi specifici. Questa soluzione funziona al meglio come supplemento al ridimensionamento automatico basato sugli obiettivi.

Puoi gestire il ridimensionamento automatico basato sul target utilizzando la GameLift console Amazon, AWS Command Line Interface (AWS CLI) o un SDK. AWS Puoi gestire il ridimensionamento automatico basato su regole utilizzando solo AWS CLI o un AWS SDK, sebbene tu possa visualizzare le politiche di scalabilità basate su regole nella console.

Argomenti

- [Scalabilità automatica basata sugli obiettivi](#)
- [Scalabilità automatica con politiche basate su regole](#)

Scalabilità automatica basata sugli obiettivi

La scalabilità automatica basata sugli obiettivi per Amazon GameLift regola i livelli di capacità in base alla metrica della flotta. `PercentAvailableGameSessions` Questa metrica rappresenta il buffer disponibile della flotta per gli aumenti improvvisi della domanda di giocatori.

Il motivo principale per mantenere un buffer di capacità è il tempo di attesa di un giocatore. Quando le slot delle sessioni di gioco sono pronte 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 il ridimensionamento automatico basato sugli obiettivi, specifica la dimensione del buffer che desideri che la flotta gestisca. Poiché `PercentAvailableGameSessions` misura la percentuale di risorse disponibili, la dimensione effettiva del buffer è una percentuale della capacità totale della flotta. Amazon GameLift 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 il ridimensionamento automatico basato sull'obiettivo

Console

1. Apri la [GameLiftconsole Amazon](#).
2. Nel pannello di navigazione, scegli Hosting, Flotte.
3. Nella pagina Flotte, scegli il nome di una flotta attiva per aprire la pagina dei dettagli della flotta.
4. Scegli la scheda Ridimensionamento. Questa scheda mostra i parametri del dimensionamento storico del parco istanze e contiene i controlli per modificare le impostazioni di dimensionamento correnti.
5. Nella sezione Capacità di ridimensionamento, verifica che i limiti di dimensione minima e massima siano appropriati per la flotta. Con la scalabilità automatica abilitata, la capacità si regola tra questi due limiti.
6. Nella politica di ridimensionamento automatico basata su Target, scegli Modifica.
7. Nella finestra di dialogo Modifica la politica di ridimensionamento automatico 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 aggiunge una nuova politica basata su obiettivi nella politica di scalabilità automatica basata su Target.

AWS CLI

1. Impostare i limiti di capacità. Imposta i valori limite utilizzando il [update-fleet-capacity](#) comando. Per ulteriori informazioni, consulta [Imposta i limiti GameLift di capacità di Amazon](#).
2. Creare una nuova policy. Apri una finestra della riga di comando e usa il [put-scaling-policy](#) comando 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 politiche basate su regole

Le politiche di scalabilità basate su regole di Amazon GameLift forniscono un controllo preciso quando ridimensionano automaticamente la capacità di una flotta in risposta all'attività dei giocatori. Per ogni policy, puoi collegare la scalabilità a una delle diverse metriche del parco veicoli, identificare un punto di attivazione e personalizzare l'evento di scalabilità o riduzione 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 raggiunge 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 il AWS Command Line Interface (AWS CLI) con l'API del servizio [Amazon GameLift](#). Puoi visualizzare tutte le politiche attive nella GameLift console Amazon.

Per interrompere temporaneamente tutte le politiche di scalabilità per una flotta, usa il AWS CLI comando [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 [update-fleet-capacity](#) comando. Per ulteriori informazioni, consulta [Imposta i limiti GameLift di capacità di Amazon](#).
2. Creare una nuova policy. Apri una finestra della riga di comando e usa il [put-scaling-policy](#) comando 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

- Apri una finestra della riga di comando e utilizza il [delete-scaling-policy](#) comando 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 costruire una dichiarazione politica di scalabilità basata su regole, specifica sei variabili:

Se il *<nome parametro>* rimane il *<valore di soglia>* dell'*<operatore di confronto>* per il *<periodo di valutazione>*, modificare la capacità del parco istanze utilizzando il *<tipo di adeguamento>* in base al *<valore di adeguamento>*.

Ad esempio, questa dichiarazione informativa avvia un evento di espansione ogni volta che la capacità extra 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 della flotta. Per una descrizione completa delle metriche, consulta.

[GameLiftMetriche Amazon per le flotte](#)

- 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 le sessioni 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 ad Amazon GameLift come confrontare i dati delle metriche 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 ridimensionamento. 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 ridimensionamento. 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 interagisce per specificare in che modo Amazon GameLift dovrebbe 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 specificato 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 la flotta di 10 istanze, indipendentemente dalle dimensioni totali della flotta.
- **Variazione percentuale della capacità:** aumenta o diminuisce la capacità attuale di una percentuale specificata. Imposta il valore di regolazione sulla percentuale in base alla quale 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 modifica 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 dal ridimensionamento automatico con politiche 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 policy.

Con più policy, ogni policy si comporta in modo indipendente. Non è possibile controllare la sequenza degli eventi di ridimensionamento. 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 fondano a vicenda. Ad esempio, è possibile creare un ciclo infinito se si creano politiche di scalabilità verso l'alto e verso il basso che impostano la capacità oltre la soglia l'una dell'altra.

Impostare la capacità minima e massima

Ogni parco istanze ha un limite di capacità massimo e uno minimo. Questa funzione 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 tua politica di ridimensionamento automatico influisca sulla capacità come previsto, aumenta il valore massimo.

La capacità della flotta è inoltre limitata dai limiti relativi al tipo di istanza della flotta e dalle quote di servizio presenti nel parco veicoli. Account AWS Non puoi impostare un minimo e un massimo al di fuori di questi limiti e quote dell'account.

Monitorare i parametri dopo una modifica della capacità

Dopo aver modificato la capacità in risposta a una politica di scalabilità automatica, Amazon GameLift attende 10 minuti prima di rispondere ai trigger della stessa politica. Questa attesa dà ad Amazon il GameLift tempo 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 valuta la politica in base alla metrica e ne tiene traccia del periodo di valutazione, che viene riavviato dopo il verificarsi di un evento di ridimensionamento. Ciò significa che una politica di scalabilità potrebbe avviare un altro evento di ridimensionamento immediatamente dopo la fine del tempo di attesa.

Non vi è alcun tempo di attesa tra gli eventi di scalabilità avviati da diverse politiche di scalabilità automatica.

Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco

Una coda di sessioni di gioco è il meccanismo principale per elaborare le richieste di nuove sessioni di gioco e localizzare i server di gioco disponibili per ospitarle. Le code offrono vantaggi significativi agli sviluppatori di giochi e ai giocatori. Eccone alcuni:

- Le code forniscono il miglior posizionamento possibile. Durante l'elaborazione delle richieste di posizionamento delle sessioni di gioco, una coda utilizza GameLift gli algoritmi Amazon per assegnare priorità alle posizioni della coda in base a una serie di preferenze definite.
- Organizza giochi su flotte Spot a prezzi inferiori. Usa le code per ottimizzare l'uso delle flotte AWS Spot, che offrono costi di hosting significativamente inferiori. Per impostazione predefinita, le code cercano sempre di inserire nuove sessioni di gioco nelle flotte Spot.
- Posiziona nuovi giochi più velocemente nei momenti in cui la domanda è elevata. Le code utilizzano più posizioni possibili per i posizionamenti. Ciò significa che è sempre disponibile una capacità di riserva se la posizione di posizionamento preferita non è disponibile.
- Rendi la disponibilità del gioco più resiliente. Possono verificarsi interruzioni. Con una coda in più sedi, un rallentamento o un'interruzione non devono necessariamente influire sull'accesso dei giocatori al gioco.
- Utilizza la capacità extra del parco istanze in modo più efficiente. Per gestire imprevisti aumenti della domanda dei giocatori, le code forniscono un accesso rapido a una capacità di hosting aggiuntiva. Le postazioni della flotta in coda forniscono una capacità di backup reciproca. Le sedi aumentano o diminuiscono in base alla domanda dei giocatori.
- Ottieni metriche sui posizionamenti delle sessioni di gioco e sulle prestazioni delle code. Amazon GameLift emette metriche sulle code, tra cui statistiche sui posizionamenti riusciti e non riusciti, il numero di richieste in coda e il tempo medio che le richieste trascorrono in coda. Puoi visualizzare queste metriche nella GameLift console Amazon o in CloudWatch.

Per iniziare con le code, consulta. [Progetta una coda per le sessioni di gioco](#)

Progetta una coda per le sessioni di gioco

Questo argomento descrive come progettare una coda che offra un'esperienza di gioco con una latenza minima e che utilizzi in modo efficiente le risorse di hosting. Per ulteriori informazioni sulle code delle sessioni di gioco e sul loro funzionamento, consulta [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#)

Queste GameLift funzionalità di Amazon richiedono code:

- [Abbinamento con FlexMatch](#)
- [Usa le istanze Spot con Amazon GameLift](#)

Definisci l'ambito della tua 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 per la tua popolazione di giocatori, crea una coda separata per ogni segmento di giocatori. Esamina ogni coda per posizionare i giocatori nei server di gioco corretti. Alcuni modi comuni per definire l'ambito delle code includono:

- Per località geografiche. Quando distribuisce 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 varianti di build o script. Se hai più di una variante del tuo server di gioco, potresti supportare gruppi di giocatori che non possono partecipare alle stesse sessioni di gioco. Ad esempio, le build o gli script dei server di gioco potrebbero supportare linguaggi o tipi di dispositivi diversi.
- Per tipo di evento. Potresti creare una coda speciale per gestire le partite dei partecipanti a tornei o altri eventi speciali.

Crea una politica di latenza dei giocatori

Se le tue richieste di piazzamento includono dati sulla latenza dei giocatori, l'algoritmo trova le sessioni di gioco nelle località con la latenza media più bassa per tutti i giocatori. Posizionare le sessioni di gioco in base alla latenza media dei giocatori GameLift impedisce ad Amazon di inserire la maggior parte dei giocatori in giochi con latenza elevata. Tuttavia, Amazon colloca GameLift ancora

i giocatori con una latenza estrema. Per soddisfare questi giocatori, crea politiche di latenza dei giocatori.

Una politica di latenza dei giocatori GameLift impedisce ad Amazon di inserire una sessione di gioco richiesta ovunque i giocatori coinvolti nella richiesta riscontrino una latenza superiore al valore massimo. Le politiche sulla latenza dei giocatori possono anche GameLift impedire ad Amazon di abbinare le richieste di sessioni 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 usare [Amazon GameLift FlexMatch](#) per 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 rimanente in coda fino al timeout alla ricerca di una posizione in cui le latenze di tutti i giocatori siano inferiori a 200 millisecondi.

Create queue

Queue settings

Name

The name must be unique and have 1-128 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Timeout

Specify how long GameLift tries to place a game session before stopping.

 seconds

Must be 10-600 seconds.

 We recommend setting player latency policies, unless you're using GameLift FlexMatch. 

Player latency policies - *optional*

Add policies to help place players into games with lower latency. Use multiple policies to reduce latency requirements per policy so that each player eventually finds a match.

0 seconds left to allocate

100%

Period start

Seconds

Period end

Seconds

Max player latency

Milliseconds

Remove

Seconds

Seconds

Milliseconds

Remove

Seconds

Seconds

Milliseconds

Remove

Add policy

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 inserire i giocatori in sessioni di gioco con una latenza minima. Se stai

creando code in più sedi che utilizzano flotte di istanze Spot, segui le istruzioni in. [Tutorial: configura una coda di sessioni di gioco per le istanze Spot](#)

Un modo per creare una coda con più sedi consiste nell'aggiungere una [flotta con più sedi](#) a una coda. In questo modo, la coda può collocare sessioni di gioco in qualsiasi posizione della flotta. Puoi anche aggiungere altre flotte con diverse configurazioni o sedi di residenza per la ridondanza. Se utilizzi un parco istanze Spot con più sedi, segui le best practice e includi un parco istanze On-Demand con le stesse sedi.

L'esempio seguente descrive 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 dispone delle seguenti località Regioni AWS di collocamento: `us-east-1`, `us-east-2`, `central-1`, `eu-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. Puoi ridurre al minimo la latenza delle richieste posizionando la coda in una posizione vicina a dove hai 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 in coda. L'ordine di destinazione determina il modo GameLift in cui Amazon organizza 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 sedi, Amazon GameLift utilizza le sedi 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

Game session placement priority

The values that GameLift uses to prioritize game session placement. The default order is latency, cost, destination, and location.

- Latency**
Prioritize locations with the lowest average player latency.
- Cost**
Prioritize destinations with the lowest current hosting cost.
- Destination**
Prioritize based on the defined destination order.
- Location**
Prioritize based on the defined location order.

▼ Location order

An ordered list of locations that the queue can use for game session placement.

Location	
ca-central-1	<input type="button" value="Remove"/>
us-east-1	<input type="button" value="Remove"/>
us-east-2	<input type="button" value="Remove"/>
us-west-2	<input type="button" value="Remove"/>

Dai priorità al posizionamento delle sessioni di gioco

Amazon GameLift utilizza l'algoritmo FleetIQ per determinare dove collocare una nuova sessione di gioco in base a una serie ordinata di criteri. Puoi utilizzare l'ordine prioritario predefinito oppure personalizzare l'ordine.

Ordine di priorità predefinito

Per le richieste di piazzamento che includono i dati sulla latenza dei giocatori, FleetIQ assegna la priorità ai criteri di posizionamento delle sessioni di gioco nel seguente ordine predefinito:

1. **Latenza:** latenza media più bassa per tutti i giocatori inclusi nella richiesta.
2. **Costo:** costo di hosting più basso, se la latenza è uguale in più sedi. Il costo dell'hosting si basa principalmente su una combinazione del tipo e della posizione dell'istanza.
3. **Destinazione:** ordine di destinazione, se latenza e costo sono uguali in più località. FleetIQ assegna la priorità alle destinazioni in base all'ordine elencato nella configurazione della coda.
4. **Posizione:** ordine delle posizioni, se latenza, costo e destinazione sono uguali in più località. FleetIQ assegna la priorità alle sedi in base all'ordine elencato nella configurazione della coda.

Ordine prioritario personalizzato

Per personalizzare l'ordine di priorità di una coda nella [GameLiftconsole Amazon](#), trascina il valore di priorità nella posizione in cui desideri. Per personalizzare l'ordine di priorità di una coda utilizzando AWS Command Line Interface (AWS CLI), usa il [create-game-session-queue](#) comando con l'--priority-configuration opzione. È possibile utilizzare questo comando per creare una nuova coda o aggiornare una coda esistente.

L'algoritmo FleetIQ aggiunge qualsiasi criterio non esplicitamente menzionato alla fine dell'elenco, in base all'ordine predefinito. Se si include il criterio della posizione nella configurazione di priorità, è necessario fornire anche un elenco ordinato di sedi.

Progetta code multiple in base alle esigenze

A seconda del gioco e dei giocatori, potresti voler creare più di una coda di sessioni di gioco. Quando il tuo 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 installare server di gioco compatibili. Questo perché i giocatori che usano la coda per partecipare ai giochi devono poter giocare su uno qualsiasi dei server di gioco della coda.
- **Gruppi di giocatori diversi.** Puoi personalizzare il modo GameLift in cui Amazon organizza le 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 sul posizionamento delle sessioni di gioco. Per ulteriori informazioni, consulta [GameLiftMetriche Amazon per le code](#).

Valuta le metriche della coda

I parametri consentono di valutare le prestazioni delle code. Puoi visualizzare le metriche relative alle code nella [GameLiftconsole Amazon o in Amazon CloudWatch](#). Per un elenco e una descrizione delle metriche delle code, consulta [GameLiftMetriche Amazon per le code](#)

Le metriche delle code possono fornire informazioni su quanto segue:

- Prestazioni complessive della coda: le metriche delle code indicano la risposta di una coda alle richieste di collocamento. 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 modificare la capacità di una coda.
- Prestazioni dell'algoritmo FleetIQ: per le richieste di posizionamento che utilizzano l'algoritmo FleetIQ, le metriche mostrano la frequenza con cui l'algoritmo trova il posizionamento ideale della sessione di gioco. Il posizionamento può dare la priorità all'utilizzo di risorse con la latenza di gioco più bassa o di risorse con il costo più basso. Esistono anche metriche di errore che identificano i motivi più comuni per cui Amazon GameLift riesce a trovare un posizionamento ideale. Per ulteriori informazioni sui parametri, consulta [Monitora Amazon GameLift con Amazon CloudWatch](#).
- Posizionamenti specifici per località: per le code in più sedi, le metriche mostrano i posizionamenti riusciti per località. Per le code che utilizzano l'algoritmo FleetIQ, questi dati forniscono informazioni utili su dove si verifica l'attività dei giocatori.

Nel valutare le metriche per le prestazioni dell'algoritmo FleetIQ, considera i seguenti suggerimenti:

- Per monitorare il tasso di ricerca di un posizionamento ideale da parte della coda, utilizza la PlacementsSucceeded metrica in combinazione con le metriche FleetIQ per la latenza più bassa e il prezzo più basso.
- Per aumentare il tasso di ricerca di una posizione ideale da parte di una coda, consulta le seguenti metriche di errore:
 - Se il valore FirstChoiceOutOfCapacity è elevato, regola il ridimensionamento della capacità per le flotte della coda.

- Se la metrica `FirstChoiceNotViable` di errore è elevata, esamina le tue flotte di istanze Spot. Le flotte di istanze Spot sono considerate non valide 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 ciascuna posizione.

Procedure consigliate per le code delle sessioni di GameLift gioco di Amazon

Ecco alcune best practice che possono aiutarti a creare code di sessione di gioco efficaci per il posizionamento delle sessioni di gioco.

Procedure ottimali per le code con qualsiasi tipo di flotta

Una coda contiene un elenco di destinazioni della flotta in cui è possibile inserire nuove sessioni di gioco. Ogni flotta può avere istanze distribuite in più località geografiche. Quando si sceglie una posizione, la coda seleziona una combinazione di una flotta e una posizione della flotta. Fornisci una serie di priorità per la coda da utilizzare quando scegli un posizionamento.

Considera le seguenti linee guida e best practice:

- Aggiungi flotte nelle località che ospitano i tuoi giocatori. Puoi aggiungere flotte e alias in qualsiasi posizione disponibile. La posizione è importante se effettui posizionamenti in base alla latenza riportata dai giocatori.
- Usa alias per tutte le flotte. Assegna un alias a ogni flotta in coda e usa i nomi degli alias per impostare le destinazioni in coda.
- Usa la stessa build o lo stesso script di gioco per tutte le flotte. La coda potrebbe mettere i giocatori in sessioni di gioco su qualsiasi flotta in coda. I giocatori devono poter giocare in qualsiasi sessione di gioco su qualsiasi flotta.
- Crea flotte in almeno due sedi. Disponendo di server di gioco ospitati in almeno un'altra posizione, puoi mitigare l'impatto delle interruzioni regionali sui tuoi giocatori. È possibile ridurre le 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 assegna la priorità alle scelte di posizionamento in base a diversi elementi, incluso l'ordine dell'elenco delle destinazioni.

- Crea la coda nella stessa posizione del tuo servizio clienti. Posizionando la coda in una posizione vicina al servizio clienti, puoi ridurre al minimo la latenza di comunicazione.
- Usa flotte con più sedi. Usa la configurazione del filtro della coda per impedire alla coda di collocare le sessioni di gioco in posizioni specifiche. Puoi utilizzare almeno due flotte con più sedi con sedi diverse per mitigare l'impatto dei posizionamenti di gioco durante un'interruzione regionale.
- Utilizza la stessa impostazione del certificato TLS per tutte le flotte. I clienti di gioco che si connettono alle sessioni di gioco delle tue flotte devono disporre di protocolli di comunicazione compatibili.

Procedure consigliate per le code con flotte Spot

Se la tua coda include flotte Spot, configura una coda resiliente. Ciò consente di risparmiare sui costi con le flotte Spot, riducendo al minimo l'effetto delle interruzioni delle sessioni di gioco. Per assistenza sulla corretta creazione di flotte e code di sessioni di gioco da utilizzare con le flotte Spot, consulta [Tutorial: configura una coda di sessioni di gioco per le istanze Spot](#). Per ulteriori informazioni sulle istanze Spot, consulta [Usa le istanze Spot con Amazon GameLift](#).

Oltre alle migliori pratiche generali della sezione precedente, prendi in considerazione le seguenti best practice specifiche di 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 quando non sono necessarie e utilizzare la scalabilità automatica per aumentare la capacità On-Demand quando le flotte Spot non sono disponibili.
- Seleziona diversi tipi di istanze su più flotte Spot in un'unica posizione. Se un tipo di istanza Spot diventa temporaneamente non disponibile, l'interruzione riguarda solo un parco istanze Spot nella sede. È consigliabile scegliere tipi di istanze ampiamente disponibili e utilizzare tipi di istanze della stessa famiglia (ad esempio, m5.large, m5.xlarge, m5.2xlarge). Usa la [GameLiftconsole Amazon](#) per visualizzare i dati storici sui prezzi, ad esempio i tipi.

Crea una coda di sessioni di gioco

Le code vengono utilizzate per posizionare nuove sessioni di gioco sulle migliori risorse di hosting disponibili in più parchi istanze e regioni. Per ulteriori informazioni sulla creazione di code per il gioco, consultare [Progetta una coda per le sessioni di gioco](#).

In un client di gioco, le nuove sessioni di gioco vengono avviate con le code utilizzando richieste di posizionamento. Scopri di più sul posizionamento delle sessioni di gioco in [Crea sessioni di gioco](#).

Quando si aggiorna la destinazione della coda in una coda, è previsto un breve periodo di transizione (fino a 30 secondi) durante il quale le sessioni di gioco effettuate nelle destinazioni in coda possono comunque finire nella vecchia flotta.

Console

1. Nella [GameLiftconsole Amazon](#), nella pagina di navigazione, scegli Code.
2. Nella pagina Queues (Code), scegliere Create queue (Crea nuova coda).
3. Nella pagina Crea coda, in Impostazioni coda, procedi come segue:
 - a. In Nome, inserisci un nome per la coda.
 - b. In Timeout, inserisci il periodo in cui desideri che Amazon GameLift tenti di effettuare una sessione di gioco prima di interromperla. Amazon GameLift cerca le risorse disponibili su qualsiasi parco veicoli fino al timeout della richiesta.
 - c. (Facoltativo) Per le politiche di latenza di Player, inserisci per quanto tempo Amazon GameLift deve cercare le risorse entro la latenza massima definita. Aggiungi policy aggiuntive per ridurre gradualmente la latenza massima. Per aggiungere criteri aggiuntivi, scegli Aggiungi politica.
4. In Posizioni di posizionamento delle sessioni di gioco, seleziona le posizioni da includere nella coda. Per impostazione predefinita, sono incluse tutte le sedi. Tutte le flotte in coda devono avere la stessa configurazione del certificato. Tutte le flotte devono eseguire build di gioco compatibili con i client di gioco che utilizzano la coda.
5. In Ordine di destinazione, aggiungi una o più destinazioni alla coda.
 - a. Scegli Add destination (Aggiungi destinazione).
 - b. Seleziona la posizione in cui si trova la destinazione.
 - c. Seleziona il tipo per la tua destinazione.
 - d. Nell'elenco risultante di nomi di parchi istanze o alias, selezionare quello da aggiungere.
 - e. Se hai più destinazioni, imposta l'ordine predefinito trascinando l'icona a sei punti a sinistra della destinazione. Amazon GameLift utilizza questo ordine quando cerca nelle destinazioni le risorse disponibili per inserire una nuova sessione di gioco.
6. Per la priorità di posizionamento delle sessioni di gioco, aggiungi e trascina i valori di Latenza, Costo, Destinazione e Posizione per definire in che modo Amazon GameLift

assegna la priorità alle flotte in coda. Per ulteriori informazioni sull'assegnazione delle priorità alle flotte, vedere [Dai priorità al posizionamento delle sessioni di gioco](#)

7. Aggiungi le località all'ordine delle località e trascinale fino alla priorità che deve essere utilizzata dalla coda. Se la posizione è l'ultima priorità per il posizionamento delle sessioni di gioco, Amazon la GameLift utilizza come fattore di spareggio.
8. (Facoltativo) In Impostazioni di notifica degli eventi, procedi come segue:
 - a. Seleziona o crea un argomento SNS per ricevere notifiche di eventi relative al posizionamento. Per ulteriori informazioni sulle notifiche degli eventi, vedere [Imposta la notifica degli eventi per il posizionamento della sessione di gioco](#).
 - b. Aggiungi dati di eventi personalizzati da aggiungere agli eventi creati da questa coda.
9. (Facoltativo) Aggiungi tag. Per ulteriori informazioni sull'etichettatura, vedere Risorse per [l'etichettatura. AWS](#)
10. Seleziona Create (Crea).

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 della flotta
- Filtri per consentire solo le posizioni `in-us-east-1`, `us-east-2`, `us-west-2`, e `ca-central-1`
- Assegna priorità alle destinazioni in base al costo e quindi alle località nell'ordine definito.

```
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="LOCATION","DESTINATION",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 della flotta e degli alias chiamando uno dei due [describe-fleet-attributes](#) o [describe-alias](#) con l'ID della flotta o dell'alias.

Se la `create-game-session-queue` richiesta ha esito positivo, Amazon GameLift restituisce un [GameSessionQueue](#) oggetto con la nuova configurazione della coda. Ora puoi 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 della flotta
- Una serie di politiche sulla 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 restituisce un [GameSessionQueue](#) oggetto con la nuova configurazione della coda.

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 configurare le notifiche degli eventi per tutti i giochi con un elevato volume di attività di posizionamento.

Sono disponibili due opzioni per impostare le notifiche di eventi.

- Chiedi ad Amazon di GameLift pubblicare 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 AmazonGameLift, consulta [Eventi di collocamento delle sessioni di gioco](#).

Configurare un argomento SNS

Affinché Amazon GameLift pubblichi tutti gli eventi generati dalla coda di una sessione di gioco su un argomento, imposta il campo target della notifica su un argomento.

Per configurare un argomento SNS per la notifica GameLift degli eventi Amazon

1. Accedere alla AWS Management Console e aprire la console Amazon SNS all'indirizzo <https://console.aws.amazon.com/sns/v3/home>.
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 politica 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) Aggiungi un controllo di accesso aggiuntivo 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 o modifica una coda esistente per aggiungerlo.

Configura un argomento SNS con crittografia lato server

Con la crittografia lato server (SSE), puoi archiviare dati sensibili in argomenti crittografati. SSE protegge il contenuto dei messaggi negli argomenti di 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 configurare un argomento SNS con crittografia lato server, consulta i seguenti argomenti:

- [Creazione di una chiave](#) nella Guida per AWS Key Management Service gli sviluppatori
- [Attivazione dell'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"
    }
  }
}
```

Configurazione di EventBridge

Amazon pubblica GameLift automaticamente tutti gli eventi di posizionamento delle sessioni di gioco su EventBridge. Con EventBridge puoi impostare regole per far sì che gli eventi vengano

indirizzati agli obiettivi 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 [Cos'è Amazon EventBridge?](#) nella Guida per l'utente di Amazon.

Di seguito sono riportati alcuni esempi di `EventBridge` regole da utilizzare con le GameLift code Amazon:

Corrisponde agli eventi di tutte le GameLift code Amazon

```
{
  "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: configura una coda di sessioni di gioco per 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 rivolto agli sviluppatori di giochi che desiderano utilizzare le flotte Spot per ospitare server di gioco personalizzati o server in tempo reale.

Cosa imparerai

- Definisci il gruppo di giocatori a cui serve la coda della sessione di gioco.
- Crea un'infrastruttura della 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 ai luoghi in cui Amazon colloca le sessioni di gioco. GameLift
- 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 [Come GameLift funziona Amazon](#).
- [Integra il tuo server di gioco con Amazon GameLift](#).
- [Carica la build del tuo server di gioco](#) o lo script in tempo reale su AmazonGameLift.
- [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 sedi 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 del Nord America.

Per ulteriori informazioni, consulta [Definisci l'ambito della tua coda](#).

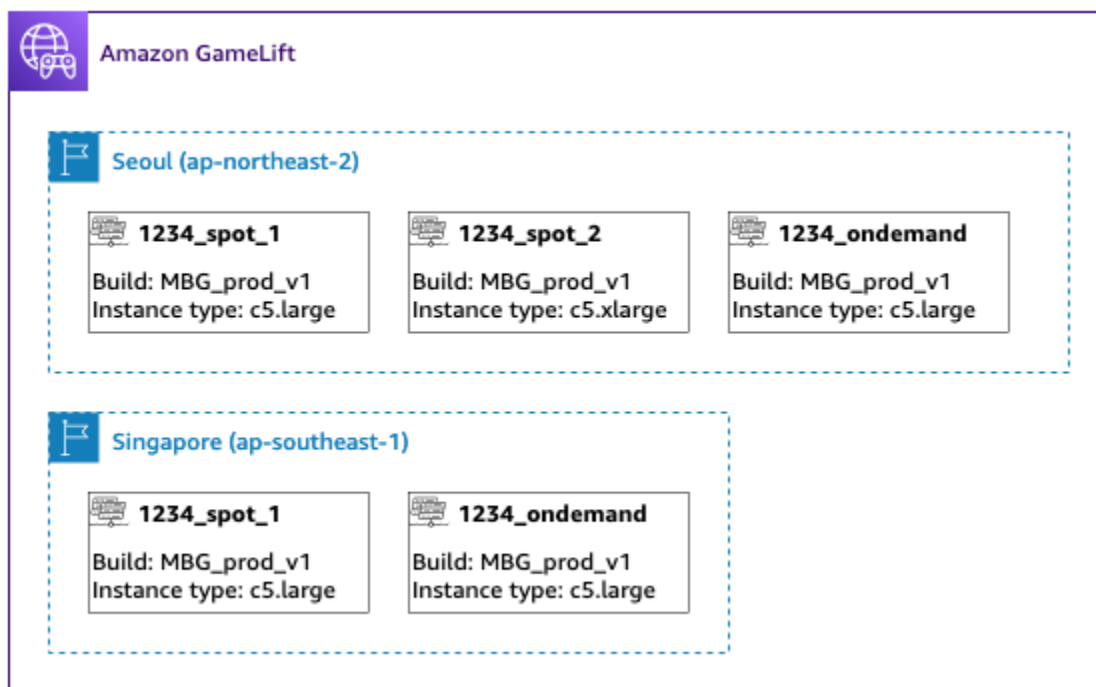
Fase 2: Creazione dell'infrastruttura della flotta Spot

Crea flotte in luoghi e con build o script di server di gioco che si adattano all'ambito che hai 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 build del server di gioco. Inoltre, prevediamo che il traffico dei giocatori aumenterà nella sede di Seoul, per cui abbiamo aggiunto altre flotte Spot.

Il diagramma seguente mostra l'esempio di infrastruttura della flotta Spot, con 3 flotte nella sede ap-northeast-2 (Seoul) e 2 flotte nella sede 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.

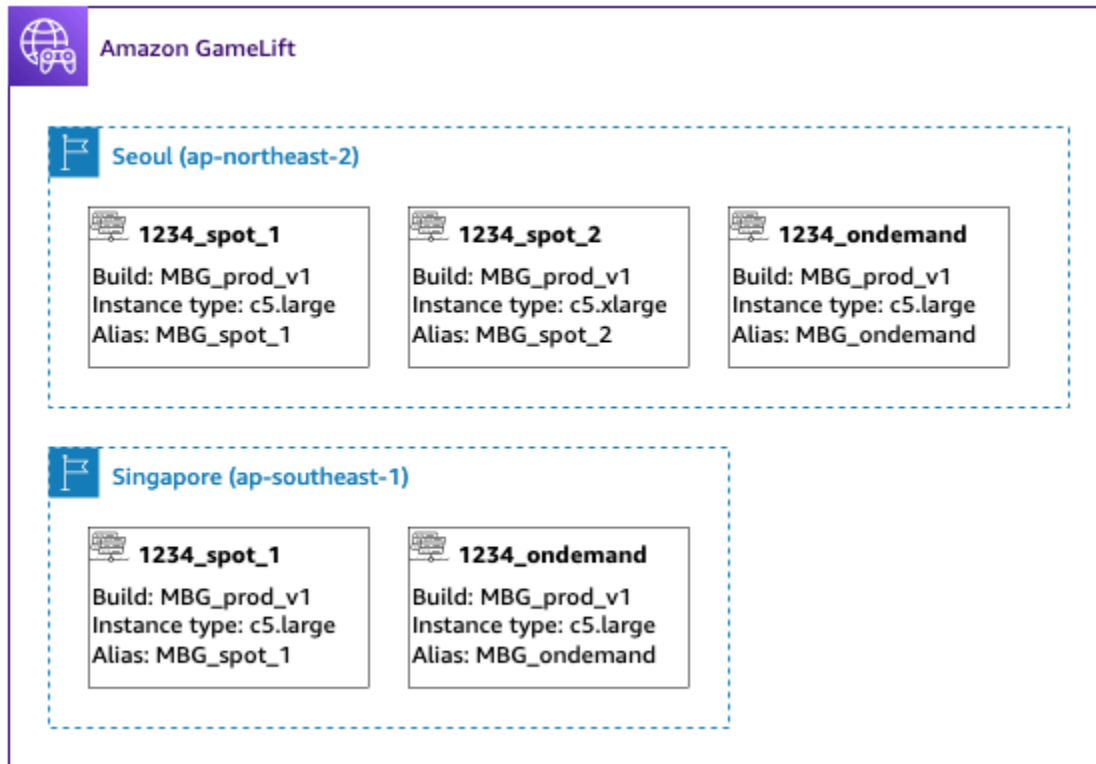


Fase 3: Assegna alias per ogni flotta

Crea un nuovo alias per ogni parco veicoli della tua infrastruttura. Gli alias astraggono le identità della flotta, rendendo efficiente la sostituzione periodica della flotta. Per ulteriori informazioni sulla creazione di alias, vedere [Aggiungi un alias a una flotta Amazon GameLift](#).

La nostra infrastruttura di flotta ha cinque flotte, quindi creiamo cinque alias utilizzando la strategia di routing. Abbiamo bisogno di tre alias nella sede Asia Pacifico (Seoul) e due alias nella sede Asia Pacifico (Singapore).

Il diagramma seguente mostra l'infrastruttura della flotta Spot descritta nella fase due 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: Crea 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, vedere. [Crea una coda di sessioni di gioco](#)

Quando crei la tua coda:

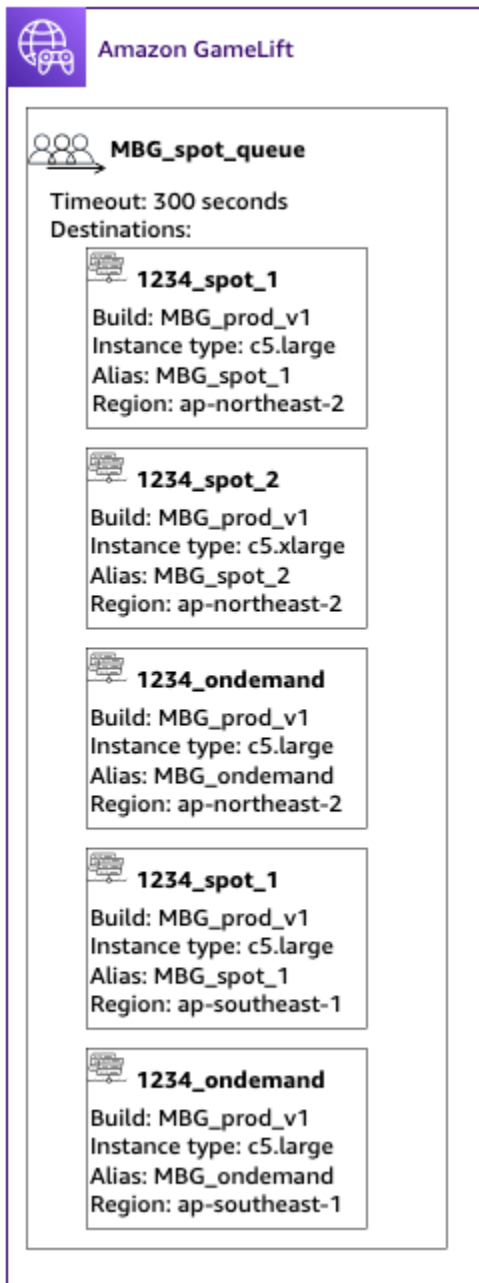
- Imposta il timeout predefinito su 10 minuti. Successivamente, potrai verificare in che modo il timeout della coda influisce sui tempi di attesa dei giocatori per iniziare le partite.
- Per ora salta la sezione sulle politiche di latenza dei giocatori. Ne parleremo nella fase successiva.
- Dai priorità alle flotte in coda. Quando si lavora con 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 priorità alle flotte prima per posizione e poi per tipo di flotta.

- Se la tua infrastruttura utilizza più sedi allo stesso modo, dai priorità alle flotte in base al tipo di flotta, posizionando le flotte Spot in cima alla coda.

Per questo tutorial, creiamo una nuova coda con il nome **MBG_spot_queue** e aggiungiamo gli alias di tutte e cinque le nostre flotte. Quindi diamo priorità ai posizionamenti prima per posizione e in secondo luogo per tipo di flotta.

In base a questa configurazione, questa coda tenta sempre di inserire nuove sessioni di gioco in una flotta Spot a Seoul. Quando queste flotte sono piene, la coda utilizza la capacità disponibile sulla flotta Seoul On-Demand come riserva. Se tutte e tre le flotte di Seoul non sono disponibili, Amazon GameLift inserisce 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.



Amazon GameLift

MBG_spot_queue

Timeout: 300 seconds
Destinations:

- 1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-northeast-2
- 1234_spot_2**
Build: MBG_prod_v1
Instance type: c5.xlarge
Alias: MBG_spot_2
Region: ap-northeast-2
- 1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-northeast-2
- 1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-southeast-1
- 1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-southeast-1

Fase 5: Aggiungere limiti di latenza alla coda

Il nostro gioco include informazioni sulla latenza nelle richieste di collocamento delle sessioni di gioco. Abbiamo anche una funzione Player Party che crea una sessione di gioco per un gruppo di giocatori. Possiamo far aspettare ancora un po' i giocatori prima di iniziare a giocare con l'esperienza di gioco ideale. I nostri test di gioco mostrano le seguenti osservazioni:

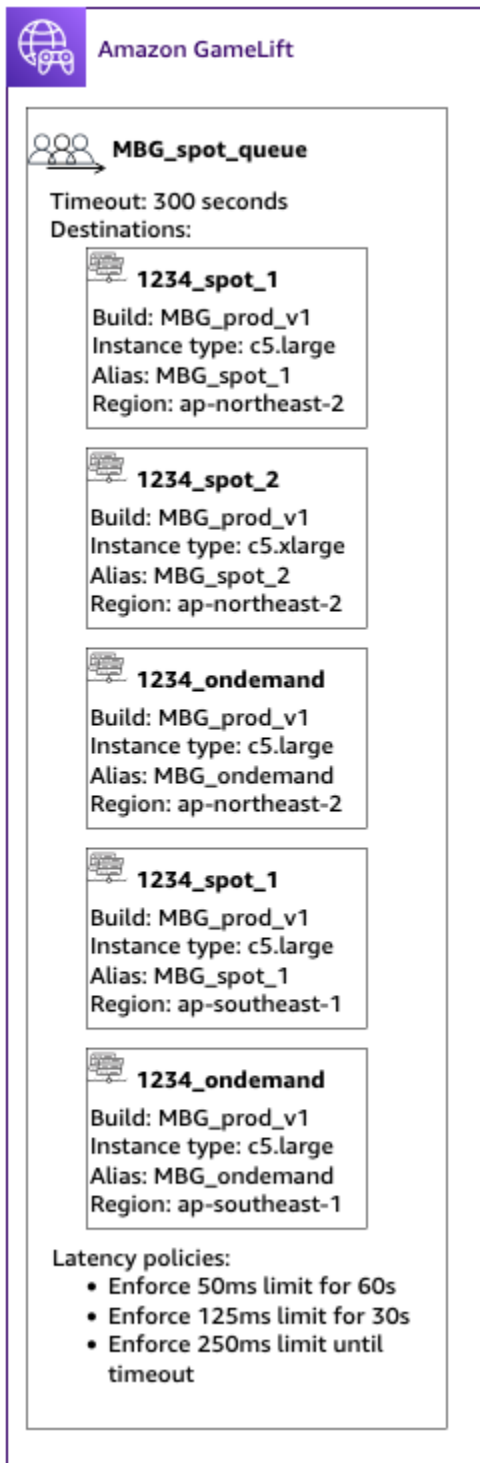
- La latenza inferiore a 50 millisecondi è ideale.

- Il gioco è ingiocabile a latenze superiori a 250 millisecondi.
- I giocatori diventano impazienti dopo circa un minuto.

Per la nostra coda, con un timeout di 300 secondi, aggiungiamo dichiarazioni politiche che limitano la latenza consentita. Le dichiarazioni politiche consentono gradualmente valori di latenza più elevati fino a 250 millisecondi di latenza.

Con questa politica, la nostra coda cerca i posizionamenti con una latenza ideale (inferiore a 50 millisecondi) per il primo minuto, quindi allenta il limite. La coda non effettua posizionamenti in cui la latenza del giocatore è di 250 millisecondi o superiore.

Il diagramma seguente mostra la coda della fase quattro con l'aggiunta delle politiche di latenza dei giocatori. Le politiche di latenza del giocatore stabiliscono, applicano il limite di 50 ms per 60 ms, impongono il limite di 125 ms per 30 secondi e il limite di 250 ms fino al timeout.



Amazon GameLift

MBG_spot_queue

Timeout: 300 seconds
Destinations:

- 1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-northeast-2
- 1234_spot_2**
Build: MBG_prod_v1
Instance type: c5.xlarge
Alias: MBG_spot_2
Region: ap-northeast-2
- 1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-northeast-2
- 1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-southeast-1
- 1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-southeast-1

Latency policies:

- Enforce 50ms limit for 60s
- Enforce 125ms limit for 30s
- Enforce 250ms limit until timeout

Riepilogo

Complimenti! Ecco le cose che hai realizzato:

- Hai una coda di sessione di gioco limitata per un segmento della tua popolazione di giocatori.

- La tua coda utilizza le flotte Spot in modo efficace ed è resiliente quando si verificano interruzioni di Spot.
- La tua coda dà la priorità alle flotte per garantire l'esperienza dei migliori giocatori.
- La coda ha dei limiti di latenza per proteggere i giocatori da brutte esperienze di gioco.

Ora puoi usare la coda per inserire sessioni di gioco per i giocatori che serve. Quando effettui richieste di collocamento di una sessione di gioco per questi giocatori, fai riferimento al nome della coda di questa sessione di gioco nella richiesta. Per ulteriori informazioni su come effettuare richieste di collocamento delle sessioni di gioco, consulta [Crea sessioni di gioco](#), o [Integrazione di un client di gioco per server in tempo reale](#).

Passaggi successivi:

- [Progetta la tua coda](#).
- [Crea una coda](#).
- [Usa una coda con il tuo client di gioco](#).

Gestisci le risorse utilizzando AWS CloudFormation

Puoi utilizzarlo AWS CloudFormation per gestire le tue GameLift risorse Amazon. In AWS CloudFormation, crei un modello che modella ogni risorsa e quindi utilizzi il modello per creare le risorse. Per aggiornare le risorse, puoi apportare le modifiche al modello e utilizzare AWS CloudFormation per implementare gli aggiornamenti. È possibile organizzare le risorse in gruppi logici, chiamati stack e set di stack.

L'utilizzo AWS CloudFormation per la manutenzione delle risorse di GameLift hosting Amazon offre 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 distribuisce 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 su AWS CloudFormation, consulta la [Guida per l'utente di AWS CloudFormation](#). Per visualizzare le informazioni sui modelli per GameLift le risorse Amazon, consulta il [riferimento al tipo di GameLift risorsa Amazon](#).

Best practice

Per una guida dettagliata sull'utilizzo AWS CloudFormation, consulta le [AWS CloudFormation best practice](#) nella Guida per l'AWS CloudFormation utente. Inoltre, queste best practice hanno una rilevanza speciale per Amazon GameLift.

- Gestisci costantemente le tue risorse tramite AWS CloudFormation. Se modifichi le tue risorse all'esterno, le AWS CloudFormation tue risorse non saranno sincronizzate con i tuoi modelli di risorse.
- Utilizza gli stack e i set di stack AWS CloudFormation 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 rimandino alle nuove flotte. Per ulteriori informazioni, consulta [Lavorare con gli stack](#) 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, vedere [Utilizzo dei set di stack](#) nella Guida per l'AWS CloudFormation utente.
- 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, una flotta con istanze Spot e una flotta con istanze su richiesta.
- 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 code e configurazioni di matchmaking tendono a ricevere un volume elevato di richieste da fonti specifiche. Posizionando le risorse vicino alla fonte di tali richieste, è possibile ridurre al minimo i tempi di percorrenza delle richieste e 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.

Usare gli AWS CloudFormation stack

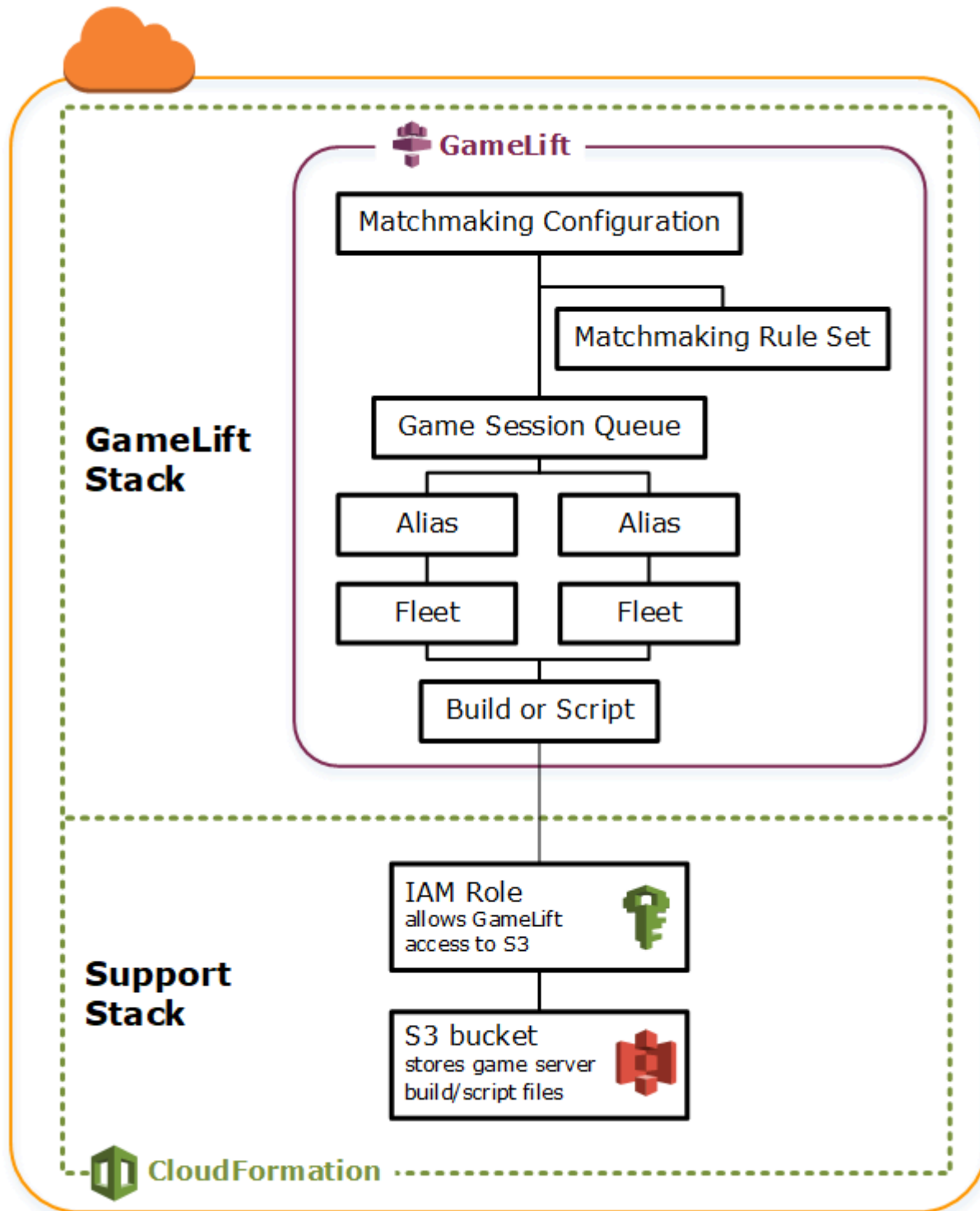
Consigliamo le seguenti strutture da utilizzare per configurare gli AWS CloudFormation stack per GameLift le risorse Amazon. La struttura ottimale degli stack varia a seconda che tu stia distribuendo il gioco in una o più località.

Pile per una singola posizione

Per gestire GameLift le risorse Amazon in un'unica posizione, consigliamo una struttura a due stack:

- **Stack di supporto:** questo stack contiene risorse da cui dipendono le tue GameLift risorse Amazon. 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 GameLift consenta ad Amazon di recuperare i tuoi file dal bucket S3 durante la creazione di una GameLift build o di una risorsa di script Amazon. Questo stack potrebbe contenere anche altre AWS risorse utilizzate nel gioco, come tabelle DynamoDB, cluster Amazon Redshift e funzioni Lambda.
- **Amazon GameLift stack:** questo stack contiene tutte le tue GameLift risorse Amazon, 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 build o script con file archiviati nella posizione del bucket S3 e distribuisce la build o lo script su una o più risorse del parco veicoli. 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 usi 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 stack per la distribuzione delle risorse in una singola regione AWS.



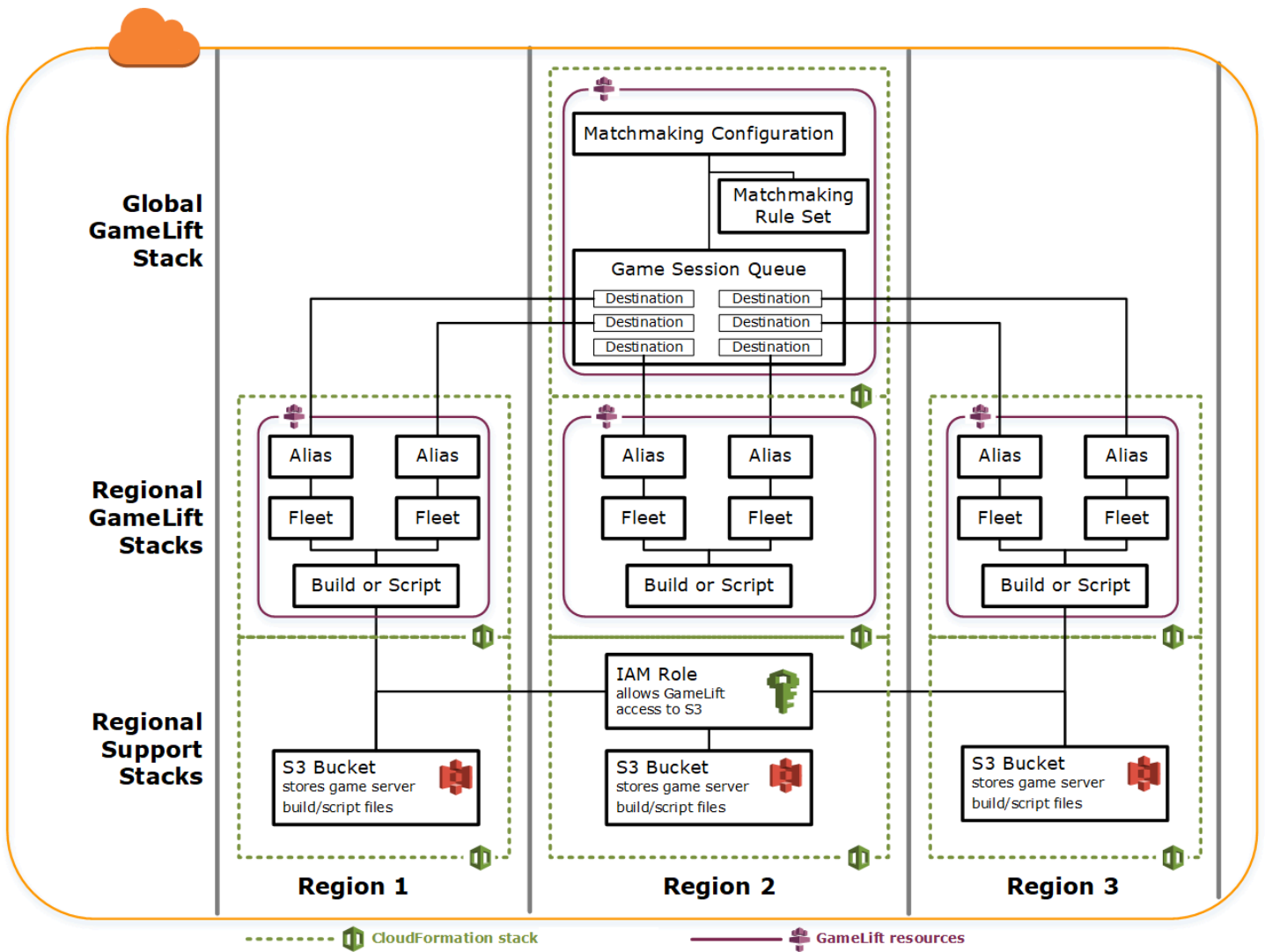
Pile per più regioni

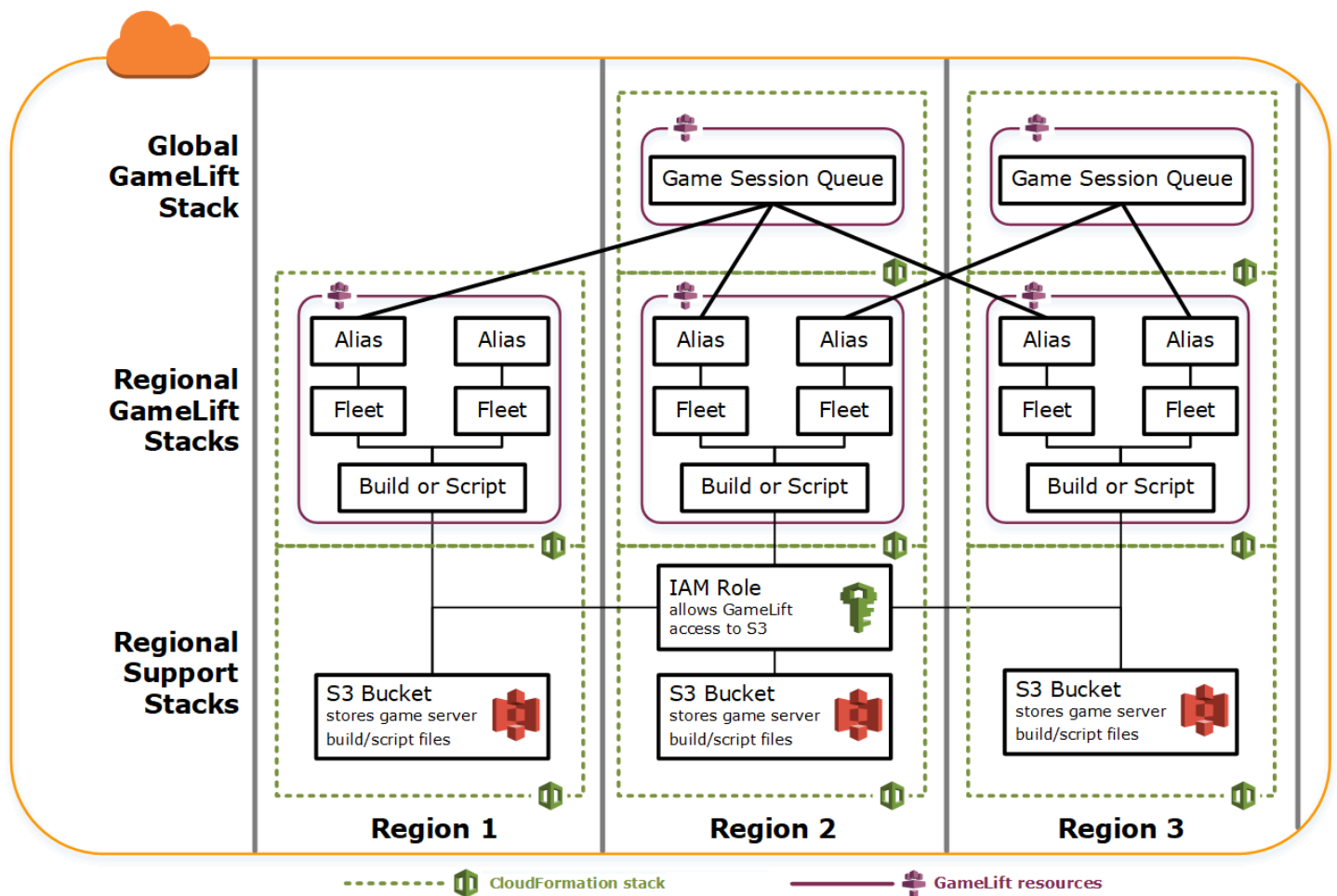
Quando distribuisce il tuo gioco in più di una regione, tieni presente come le risorse possono interagire tra le regioni. Alcune risorse, come le GameLift flotte Amazon, possono fare riferimento solo ad altre

risorse nella stessa regione. Altre risorse, come una GameLift coda Amazon, sono indipendenti dalla regione. Per gestire GameLift le risorse Amazon in più regioni, consigliamo la seguente struttura.

- **Stack di supporto regionali:** questi stack contengono risorse da cui dipendono le tue GameLift risorse Amazon. Questo stack deve includere il bucket S3 in cui si memorizzano il server di gioco personalizzato o i file di script Realtime. Potrebbe anche contenere altre AWS risorse per il tuo gioco, come tabelle DynamoDB, cluster Amazon Redshift e funzioni Lambda. Molte di queste risorse sono specifiche della regione, quindi è necessario crearle in ogni regione. Amazon necessita GameLift anche di un ruolo IAM che consenta l'accesso a queste risorse di supporto. Poiché un ruolo IAM è indipendente dalla regione, è necessaria solo una risorsa di ruolo, collocata in qualsiasi regione e referenziata in tutti gli altri stack di supporto.
- **GameLiftStack Amazon regionali:** questo stack contiene le GameLift risorse Amazon 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 build o script con file in una posizione del bucket S3 e distribuisce la build o lo script su una o più risorse del parco veicoli. 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.
- **Stack Amazon globale:** questo GameLift 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.

Nei diagrammi riportati di seguito viene illustrata una struttura a più stack 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.





Aggiornamento delle build

Le GameLift build di Amazon 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'AWS CloudFormationutente](#).

Distribuisce automaticamente gli aggiornamenti delle build

Quando aggiorni uno stack contenente risorse build, parco istanze e alias correlate, il comportamento predefinito di AWS CloudFormation consiste nell'eseguire automaticamente questi passaggi in sequenza. Attivi questo aggiornamento caricando prima i nuovi file della build in un nuovo percorso S3. Quindi modifichi il modello di build AWS CloudFormation per puntare al nuovo percorso 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 crea una nuova build Amazon. GameLift
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 anche possibile mantenere le risorse AWS CloudFormation invece di eliminarle. Per ulteriori informazioni, consulta [RetainResources](#) nella documentazione di riferimento dell'API AWS CloudFormation.

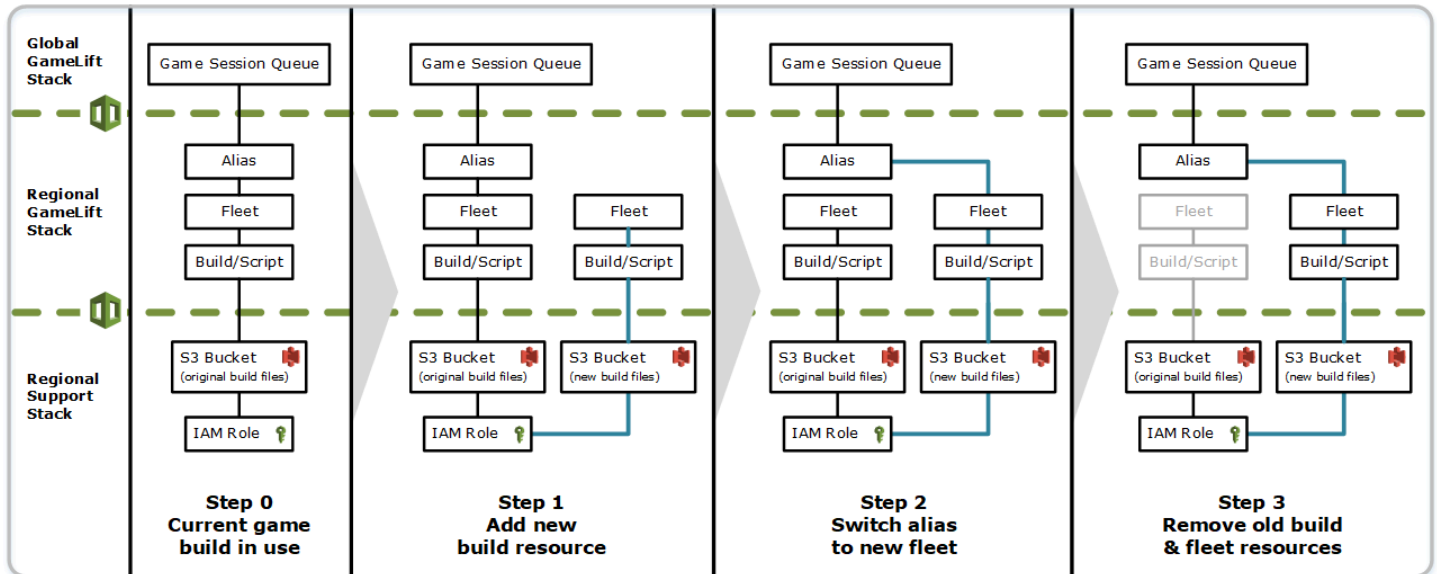
Distribuisce gli aggiornamenti delle build manualmente

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 la GameLift console Amazon o l'interfaccia a riga di comando. 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 si aggiorna il modello, AWS CloudFormation crea una seconda

risorsa di build e i parchi istanze corrispondenti. Poiché le risorse esistenti non vengono sostituite, non vengono cancellate e gli alias continuano a puntare i parchi istanze 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. Vengono create una nuova GameLift build e una nuova flotta Amazon 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 non riesce o si blocca, consulta [Continuare a ripristinare un aggiornamento](#) nella Guida per l'AWS CloudFormation utente.

Peering VPC per Amazon GameLift

Questo argomento fornisce indicazioni su come configurare una connessione peering VPC tra i server di gioco GameLift ospitati su Amazon e le altre risorse non Amazon. GameLift Usa le connessioni peering di Amazon Virtual Private Cloud (VPC) per consentire ai tuoi server di gioco di comunicare direttamente e privatamente con AWS le altre 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 ulteriori 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 Amazon VPC e VPC peering, tieni presente che la configurazione del peering con i server di GameLift gioco Amazon è leggermente diversa. Non hai accesso al cloud privato virtuale che contiene i tuoi server di gioco, che è controllato dal GameLift servizio Amazon, quindi non puoi richiederne direttamente il peering. Invece, devi prima pre-autorizzare il VPC con le tue GameLift risorse non Amazon ad accettare una richiesta di peering dal servizio Amazon. GameLift Quindi fai in modo che Amazon GameLift richieda il peering VPC che hai appena autorizzato. Amazon GameLift gestisce le attività di creazione della connessione peering, impostazione delle tabelle dei percorsi e configurazione della connessione.

Configurazione del peering VPC per un parco istanze esistente

1. Ottieni ID e credenziali dell'AWSaccount.

Hai bisogno di un ID e delle credenziali di accesso per i seguenti AWS account. Puoi trovare gli ID AWS dell'account accedendo [AWS Management Console](#) e visualizzando le impostazioni del tuo account. Per ottenere le credenziali, andare alla console IAM.

- AWSaccount che usi per gestire i tuoi server di GameLift gioco Amazon.
- AWSaccount che usi per gestire le tue GameLift risorse non Amazon.

Se utilizzi lo stesso account per GameLift risorse Amazon GameLift e non Amazon, ti servono ID e credenziali solo per quell'account.

2. Ottenimento di identificatori per ogni VPC.

Ricevere le informazioni riportate di seguito per i due VPC che devono essere collegati in peer:

- VPC per i tuoi server di GameLift gioco Amazon: questo è l'ID del tuo GameLift parco veicoli Amazon. I tuoi server di gioco sono distribuiti in Amazon GameLift su una flotta di istanze EC2. Una flotta viene automaticamente inserita nel proprio VPC, gestito dal GameLift servizio Amazon. Non hai accesso diretto al VPC, quindi è identificato dall'ID della flotta.
- VPC per le tue GameLift AWS risorse non Amazon: puoi creare un peering VPC con qualsiasi risorsa in esecuzione AWS e gestita da un AWS account a cui hai accesso. Se non hai già creato un VPC per queste risorse, consulta [Nozioni di base su Amazon VPC](#). Una volta creato un VPC, è possibile individuare l'ID VPC accedendo a [AWS Management Console](#) per Amazon VPC e visualizzare i VPC.

Note

Al momento della configurazione di un peering, entrambi i VPC devono esistere nella stessa regione. Il VPC per i server di gioco GameLift della tua flotta Amazon si trova nella stessa regione della flotta.

3. Autorizza un peering VPC.

In questo passaggio, stai pre-autorizzando una futura richiesta di Amazon GameLift per collegare il cloud privato virtuale ai tuoi server di gioco e al tuo cloud privato virtuale per risorse non Amazon. GameLift Questa operazione aggiorna il gruppo di sicurezza per il VPC.

Per autorizzare il peering VPC, chiama l'API del GameLift servizio Amazon [CreateVpcPeeringAuthorization\(\)](#) o usa il comando CLI AWS. `create-vpc-peering-authorization` Effettua questa chiamata utilizzando l'account che gestisce le tue GameLift risorse non Amazon. Identificare le informazioni che seguono:

- Peer VPC ID: si tratta del VPC con risorse non Amazon. GameLift
- ID GameLift AWS account Amazon: questo è l'account che utilizzi per gestire la tua GameLift flotta Amazon.

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\(\)](#) (AWSCLI `describe-vpc-peering-authorizations`).
- [DeleteVpcPeeringAuthorization\(\)](#) (AWSCLI `delete-vpc-peering-authorization`).

4. Richiedi una connessione peering.

Con un'autorizzazione valida, puoi richiedere ad Amazon di GameLift stabilire una connessione peering.

Per richiedere un peering VPC, chiama l'API del GameLift servizio Amazon [CreateVpcPeeringConnection\(\)](#) o usa il comando AWS CLI. `create-vpc-peering-connection` Effettua questa chiamata utilizzando l'account che gestisce i tuoi server di GameLift gioco Amazon. Utilizza le seguenti informazioni per identificare i due VPC da sottoporre al peering:

- Peer VPC ID e ID AWS account: si tratta del VPC per le tue GameLift risorse non Amazon 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 i tuoi server di GameLift gioco Amazon.

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 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\(\)](#) (AWSCLIdescribe-vpc-peering-connections).
- [DeleteVpcPeeringConnection\(\)](#) (AWSCLIdelete-vpc-peering-connection).

Per configurare il peering VPC con un nuovo parco istanze

Puoi creare una nuova GameLift flotta Amazon e richiedere contemporaneamente una connessione peering VPC.

1. Ottieni ID e credenziali dell'AWSaccount.

Hai bisogno di un ID e delle credenziali di accesso per i due AWS account seguenti. Puoi trovare gli ID AWS dell'account accedendo [AWS Management Console](#) e visualizzando le impostazioni del tuo account. Per ottenere le credenziali, andare alla console IAM.

- AWSaccount che usi per gestire i tuoi server di GameLift gioco Amazon.
- AWSaccount che usi per gestire le tue GameLift risorse non Amazon.

Se utilizzi lo stesso account per GameLift risorse Amazon GameLift e non Amazon, ti servono ID e credenziali solo per quell'account.

2. Ottieni l'ID VPC per le tue risorse non Amazon GameLiftAWS.

Se non hai già 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 le tue GameLift risorse non Amazon sono gestite con un AWS account o un gruppo di utenti/utenti diverso da quello che usi con AmazonGameLift, dovrai utilizzare queste credenziali dell'account per richiedere l'autorizzazione nella fase successiva.

Una volta creato un VPC, puoi individuare l'ID VPC nella console Amazon VPC visualizzando i VPC.

3. Autorizza un peering VPC con risorse non Amazon. GameLift

Quando Amazon GameLift crea la nuova flotta e il VPC corrispondente, invia anche una richiesta di peer with the VPC per le tue risorse non Amazon. GameLift Devi preautorizzare la richiesta. Questa fase aggiorna il gruppo di sicurezza per il VPC.

Utilizzando le credenziali dell'account che gestiscono le tue GameLift risorse non Amazon, chiama l'API del GameLift servizio Amazon [CreateVpcPeeringAuthorization\(\)](#) o usa il comando AWS CLI. `create-vpc-peering-authorization` Identificare le informazioni che seguono:

- Peer VPC ID: ID del cloud privato virtuale con le tue risorse non Amazon. GameLift
- ID GameLift AWS account Amazon: ID dell'account che utilizzi per gestire la tua GameLift flotta Amazon.

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\(\)](#) (AWSCLI `describe-vpc-peering-authorizations`).
 - [DeleteVpcPeeringAuthorization\(\)](#) (AWSCLI `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 utilizzi per gestire il VPC con le tue risorse non GameLift Amazon.
 - `peer-vpc-id` — ID del VPC con il tuo GameLift account esterno.

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`. Puoi monitorare il successo o il fallimento della richiesta di peering [describe-vpc-peering-connections](#) chiamando.

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 è identificato in modo univoco dalla combinazione dell'ID del tuo GameLift AWS account non Amazon e dell'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 i tuoi server di GameLift gioco Amazon, considera queste cause principali comuni:

- Non è stata trovata alcuna autorizzazione per la connessione richiesta:
 - Verifica lo stato di un'autorizzazione VPC per il VPC non GameLift Amazon. Potrebbe non esistere o essere scaduto.
 - Controlla le regioni dei due VPC che stai cercando di sottoporre al peering. 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 VPC si sovrappongono. I blocchi CIDR IPv4 assegnati a VPC con connessione peer non si possono sovrapporre. Il blocco CIDR del VPC per la tua GameLift flotta Amazon viene assegnato automaticamente e non può essere modificato, quindi dovrai modificare il blocco CIDR del VPC per le tue risorse non Amazon. GameLift Per risolvere il problema:
 - Cerca questo blocco CIDR per la tua GameLift flotta Amazon `DescribeVpcPeeringConnections()` chiamando.
 - Vai alla console Amazon VPC, trova il VPC per le tue GameLift risorse non Amazon 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.

Visualizzazione dei dati di gioco nella console

Il GameLift servizio gestito di Amazon raccoglie continuamente dati relativi ai giochi attivi per aiutarti a comprendere il comportamento e le prestazioni dei giocatori. Con la GameLift console Amazon, puoi visualizzare, gestire e analizzare queste informazioni per le tue build, flotte, sessioni di gioco e sessioni dei giocatori.

Argomenti

- [Visualizza il tuo GameLift stato attuale su Amazon](#)
- [Visualizza le tue build](#)
- [Visualizza i tuoi script](#)
- [Visualizza le tue flotte](#)
- [Visualizza i dettagli della flotta](#)
- [Visualizza i dati sulle sessioni di gioco e dei giocatori](#)
- [Visualizza i tuoi alias](#)
- [Visualizza le tue code](#)

Visualizza il tuo GameLift stato attuale su Amazon

La GameLift dashboard di Amazon 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 la GameLift dashboard di Amazon

- Nella [GameLift console Amazon](#), nel pannello 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.

☰
GameLift > Dashboard

Dashboard

Build status overview View builds

Viewing data for all builds in N. Virginia region.

✔ Ready
1

⊖ Initialized
0

✘ Failed
0

Fleet status overview View fleets

Viewing data for all fleets in N. Virginia region.

✔ Active
0

⊖ Deleting
0

⊖ In progress
0

⊖ New
0

✘ Error
0

⊖ Terminated
0

Resources (1) View service quotas

Resource type	Count
Builds	1
Scripts	0
Fleets	0
Aliases	0
Queues	0
Matchmaking rule sets	0
Matchmaking configurations	0

Prepare for your game launch [Learn more](#)

Fill out a launch questionnaire

Fill out our game launch questionnaire and email it to the GameLift launch team to ensure a smooth launch. The GameLift launch team will verify your GameLift setup and service limits, preparing you for launch.

[Prepare to launch](#)

Features spotlight

Updates on features available in N. Virginia region

March 22, 2022

Updates to Amazon GameLift FlexMatch for greater flexibility

October 28, 2021

New Asia Pacific (Osaka) region and Graviton2 support for Amazon GameLift

Visualizza le tue build

Nella pagina Builds della [GameLiftconsole Amazon](#), puoi visualizzare le informazioni e gestire tutte le build dei server di gioco che hai caricato su Amazon. GameLift Nel riquadro di navigazione, scegli Hosting, Builds.

La pagina Builds mostra le seguenti informazioni per ogni build:

Note

La pagina Costruzioni mostra solo le build nella tua AWS regione attuale.

- Nome: il nome associato alla build caricata.
- Stato: lo stato della build. Visualizza uno dei tre messaggi di stato:
 - Inizializzato: il caricamento non è stato avviato o è ancora in corso.
 - Pronta: la build è pronta per la creazione della flotta.
 - Fallito: la compilazione è scaduta prima che Amazon GameLift ricevesse i file binari.
- Ora di creazione: la data e l'ora in cui hai caricato la build su AmazonGameLift.
- 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 build determina quale sistema operativo Amazon GameLift installa sulle istanze di una flotta.
- Dimensione: la dimensione, in megabyte (MB), del file di build caricato su Amazon. GameLift
- Flotte: il numero di flotte dispiegate con 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 costruzione

Nella pagina Costruzioni, 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 Costruzioni. La sezione Flotte mostra un elenco di flotte create con la build, incluse le stesse informazioni di riepilogo della pagina [Flotte](#).

Visualizza i tuoi script

Nella pagina Script della [GameLiftconsole Amazon](#), puoi visualizzare le informazioni e gestire tutti gli script di Realtime Servers che hai caricato su Amazon. GameLift Nel riquadro di navigazione, scegli Hosting, Script.

La pagina Script mostra le seguenti informazioni per ogni script:

Note

La pagina Script mostra solo gli script nella tua AWS regione corrente.

- Nome: il nome associato allo script caricato.
- ID: l'ID univoco assegnato allo script al momento del caricamento.
- Versione: l'etichetta della versione associata allo script caricato.
- Dimensione: la dimensione, in megabyte (MB), del file di script caricato su Amazon. GameLift
- Ora di creazione: la data e l'ora in cui hai caricato lo script su AmazonGameLift.
- Flotte: il numero di flotte distribuite 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.
- Filtrare e ordinare 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](#).

Visualizza le tue flotte

Puoi visualizzare le informazioni su tutte le flotte create per ospitare i tuoi giochi su Amazon nel GameLift 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 di flotta o eliminare un parco veicoli.

Per visualizzare la pagina Flotte, scegli Flotte dal riquadro di navigazione.

La pagina Parchi istanze per impostazione predefinita contiene le seguenti informazioni riepilogative. Puoi personalizzare le informazioni visualizzate scegliendo il pulsante Impostazioni (ingranaggio).

- Nome: nome amichevole 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: la data e l'ora di creazione della flotta.
- Tipo di elaborazione: il tipo di elaborazione utilizzato per ospitare i giochi. Una flotta può essere una flotta Managed EC2 o una flotta Anywhere.
- Tipo di istanza: il tipo di istanza Amazon EC2, che determina la capacità di calcolo delle istanze del parco istanze.
- Istanze attive: il numero di istanze EC2 in uso per il parco veicoli.
- Istanze desiderate: il numero di istanze EC2 da mantenere attive.
- Sessioni di gioco: il numero di sessioni di gioco attive in corso nella flotta. I dati vengono posticipati di cinque minuti.

Visualizza i dettagli della flotta

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 esecuzione di una flotta.
- Aggiungi o rimuovi le sedi della flotta.
- Modificare le impostazioni della capacità del parco istanze.
- Imposta o modifica il ridimensionamento automatico del tracciamento degli obiettivi.
- Eliminare un parco istanze.

Informazioni

Impostazioni della flotta

- Fleet ID: identificatore univoco assegnato alla flotta.
- Nome: il nome della flotta.
- ARN: l'identificatore assegnato a questa flotta. L'ARN di una flotta la identifica come GameLift risorsa Amazon e specifica la regione e l'account. AWS
- Descrizione: breve descrizione identificabile della flotta.
- Stato: stato attuale della flotta, che può essere Nuovo, In download, In costruzione e Attivo.
- Ora di creazione: la data e l'ora in cui è stata creata la 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 veicoli utilizza istanze on-demand o spot.
- Tipo EC2: tipo di [istanza](#) Amazon EC2 selezionato per il parco veicoli 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: se la flotta è abilitata o disattivata, è possibile utilizzare un certificato TLS per autenticare un server di gioco e crittografare tutte le comunicazioni client/server.
- Gruppo metrico: il gruppo utilizzato per aggregare le metriche per più flotte.

- **Politica di protezione contro la scalabilità del gioco:** 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 della Politica.
- **Periodo di validità:** quanto tempo occorre attendere prima di reimpostare il numero di sessioni create da un giocatore.

Dettagli della costruzione

La sezione Dettagli della build mostra la build ospitata nella flotta. Seleziona il nome della build per visualizzare la pagina completa dei dettagli della build.

Configurazione del runtime

La sezione Runtime configuration 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 il tempo di attesa affinché il processo si attivi prima di terminarlo.

Impostazioni porta EC2

La sezione Porte mostra le autorizzazioni di connessione della flotta, inclusi gli indirizzi IP e gli intervalli di impostazione delle porte.

Parametri

Nella scheda Parametri è visualizzata una rappresentazione grafica dei parametri del parco istanze nel tempo. Per ulteriori informazioni sull'utilizzo delle metriche in AmazonGameLift, consulta [Monitora Amazon GameLift 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. Consulta le descrizioni degli [eventi](#) in Amazon GameLift API Reference.

Dimensionamento

La scheda Scalabilità contiene informazioni sulla capacità della flotta, incluso lo stato attuale e le modifiche della capacità nel tempo. Fornisce inoltre strumenti per aggiornare i limiti di capacità e gestire il dimensionamento automatico.

Capacità di scalabilità

Visualizza le impostazioni attuali della capacità della flotta per ogni sede della flotta. Per ulteriori informazioni sulla modifica dei limiti e della capacità, vedere [Scalabilità della capacità di GameLift hosting di Amazon](#).

- **AWSPosizione:** nome di una sede in cui vengono distribuite le istanze del parco veicoli.
- **Stato:** stato di hosting della sede della flotta. Lo stato della posizione ACTIVE deve essere tale da poter ospitare 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 ridimensionamento per avviare o chiudere le istanze in base alle necessità fino a quando le istanze attive non sono uguali alle istanze desiderate.
- **Dimensione massima:** la maggior parte delle 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 scalabilità automatica

Questa sezione contiene informazioni sulle politiche di scalabilità automatica 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'AWSSDK o l'interfaccia a riga di comando, sono visualizzate qui. Per ulteriori informazioni sulla scalabilità, vedere [Ridimensiona automaticamente la capacità della flotta con Amazon GameLift](#).

Cronologia della scalabilità

Visualizza i grafici delle variazioni di capacità nel tempo.

Locations

La scheda Posizioni elenca tutte le sedi in cui vengono distribuite le istanze del parco veicoli. Le sedi includono la regione di origine della flotta e tutte le sedi remote che sono state aggiunte. Puoi aggiungere o rimuovere sedi direttamente in questa scheda.

- **Posizione:** nome di una sede in cui vengono distribuite le istanze del parco veicoli.
- **Stato:** stato di hosting della sede della flotta. Lo stato della posizione traccia il processo di attivazione delle prime istanze nella posizione. Lo stato della posizione ACTIVE deve essere tale da poter ospitare partite.
- **Istanze attive:** il numero di istanze con processi server in esecuzione nella sede del parco veicoli.
- **Server attivi:** il numero di processi del server di gioco in grado di ospitare sessioni di gioco nella sede della flotta.
- **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 della sessione di gioco per accedere a informazioni aggiuntive sulla sessione di gioco, incluse le sessioni dei giocatori. Per ulteriori informazioni sulle sessioni dei giocatori, consulta [Visualizza i dati sulle sessioni di gioco e dei giocatori](#).

Visualizza i dati sulle sessioni di gioco e dei giocatori

Puoi visualizzare informazioni sulle sessioni di gioco e sui singoli giocatori. Per ulteriori informazioni sulle sessioni di gioco e sulle sessioni giocatore, consulta [Come i giocatori si connettono ai giochi](#).

Per visualizzare i dati della sessione di gioco e dei giocatori

1. Nella [GameLiftconsole Amazon](#), nel riquadro di navigazione, scegli Flotte.
2. Scegli la flotta dall'elenco delle flotte che ha ospitato le tue sessioni di gioco.
3. Scegli la scheda Sessioni di gioco. In questa scheda sono riportate tutte le sessioni di gioco del parco istanze, oltre a informazioni riepilogative.

4. Scegli una sessione di gioco per visualizzare informazioni aggiuntive sulla sessione di gioco e un elenco di giocatori che si sono connessi al gioco.

Informazioni

Panoramica

Questa sezione mostra un riepilogo delle informazioni sulla sessione di gioco.

- **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 della risorsa Amazon della sessione di gioco.
- **Nome:** nome generato per la sessione di gioco.
- **Posizione:** il luogo in cui Amazon GameLift ha ospitato la sessione di gioco.
- **Ora di creazione:** data e ora in cui Amazon GameLift ha creato la sessione di streaming.
- **Ora di fine:** data e ora di fine della 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.
- **ID creatore:** 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 delle sessioni di gioco da impostare su tutte le nuove istanze che Amazon GameLift avvia nel parco giochi.

Dati di gioco

Dati ben formattati da inviare alla sessione di gioco all'avvio.

Proprietà del gioco

Proprietà delle coppie di chiavi e valori che influenzano la sessione di gioco.

Dati di matchmaking

Il FlexMatch matchmaker JSON. Per rivedere e modificare il matchmaker, scegli [Visualizza la configurazione del matchmaking](#). Per ulteriori informazioni sul FlexMatch matchmaking, vedi [Costruisci un matchmaker](#).

Sessioni giocatore

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 informazioni aggiuntive sul giocatore.
- Stato: lo stato della sessione del giocatore. Di seguito sono riportati gli stati possibili:
 - 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.
 - Completato: la sessione del giocatore è terminata; il giocatore non è più connesso.
 - Timed out: il lettore 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 del giocatore: informazioni sul giocatore fornite durante la creazione della sessione del giocatore.

Informazioni sul giocatore

Visualizza ulteriori informazioni su un giocatore, fra cui un elenco di tutti i giochi a cui si è connesso in tutti i parchi istanze della regione corrente. Queste informazioni includono lo stato, l'ora di inizio, l'ora di fine e il tempo totale di connessione per ogni sessione del giocatore. Puoi scegliere di visualizzare i dati per le sessioni di gioco e le flotte pertinenti.

Visualizza i tuoi alias

La pagina Alias mostra informazioni sugli alias della flotta che hai creato nella tua regione attuale. Per visualizzare la pagina degli alias, scegli Alias nel riquadro di navigazione.

Puoi fare quanto segue nella pagina degli alias:

- Crea un nuovo alias. Scegli Crea alias.
- Filtrare e ordinare la tabella degli alias. Utilizzare i controlli nella parte superiore della tabella.
- 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 visualizza le informazioni sull'alias.

In questa pagina, è possibile:

- Modifica un alias. Scegliere Modifica.
- Visualizza le flotte che hai associato all'alias.
- Eliminare un alias. Scegliere Elimina.

Le informazioni dettagliate sugli alias includono:

- ID: il numero univoco utilizzato per identificare l'alias.
- Descrizione: la descrizione dell'alias.
- ARN: il nome della risorsa Amazon 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 di 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 una posizione in cui possono ottenere un upgrade.
- Tag: coppie di chiavi e valori utilizzate per identificare l'alias.

Visualizza le tue code

È possibile visualizzare le informazioni su tutte le code di posizionamento delle sessioni di gioco esistenti. La pagina delle code mostra le code create nella tua regione corrente. 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 [Configurazione delle GameLift code Amazon per il posizionamento delle sessioni di gioco](#).

La pagina delle code mostra le seguenti informazioni di riepilogo per ogni coda:

- 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 di tempo massimo, in secondi, in cui una richiesta di piazzamento di una sessione di gioco rimane in coda prima del timeout.
- Destinazioni in coda: numero di flotte elencate nella configurazione della coda. Amazon GameLift 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. Scegliere Modifica.
- Eliminare una coda. Dopo aver eliminato una coda, tutte le richieste di nuove sessioni di gioco che fanno riferimento a quel nome avranno esito negativo. Scegliere 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) e il timeout della coda. Puoi usare l'ARN per fare riferimento alla coda in altre azioni o aree di Amazon. GameLift Il timeout è il periodo

di tempo massimo, in secondi, durante il quale una richiesta di piazzamento di una sessione di gioco rimane in coda prima del timeout.

Notifiche di eventi

La sezione Notifica degli eventi elenca l'argomento SNS su cui Amazon GameLift pubblica le notifiche degli eventi e i dati degli 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, vedere Risorse per l'[etichettatura. AWS](#)

Parametri

Nella scheda Metrics (Parametri) è mostrata una rappresentazione grafica dei parametri della coda nel tempo.

Le metriche relative alle code includono una serie di informazioni che descrivono le attività di collocamento all'interno della 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 regionali possono aiutare a rilevare problemi con la progettazione generale della coda.

Le metriche relative alle code sono disponibili anche su Amazon. CloudWatch Per le descrizioni delle metriche disponibili, consulta. [GameLiftMetriche Amazon per le code](#)

Destinazioni

La scheda Destinations (Destinazioni) mostra tutti i parchi istanze o gli alias elencati per la coda.

Quando Amazon GameLift cerca nelle destinazioni le risorse disponibili per ospitare una nuova sessione di gioco, cerca nell'ordine predefinito elencato qui. Finché c'è spazio nella prima destinazione elencata, Amazon vi GameLift 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 indicano GameLift ad Amazon di cercare una destinazione disponibile con la latenza media dei giocatori più bassa. Per ulteriori informazioni sulla progettazione delle code, vedere. [Progetta una coda per le 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.

Locations

La sezione Posizioni mostra i luoghi in cui questa coda può inserire una sessione di gioco.

Priority (Priorità)

La sezione Priorità mostra l'ordine in cui la coda valuta i dettagli di una sessione di gioco.

Ordine della posizione

La sezione Ordine della posizione mostra l'ordine predefinito utilizzato dalla coda per effettuare le sessioni di gioco. La coda utilizza questo ordine se non hai definito altri tipi di priorità.

Monitoraggio di Amazon GameLift

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon GameLift e delle altre AWS soluzioni. Le metriche di Amazon sono utilizzate principalmente per tre scopi GameLift: 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 per monitorare Amazon GameLift, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- GameLift Console Amazon
- Amazon CloudWatch: puoi monitorare i GameLift parametri di Amazon in tempo reale, nonché i parametri di altre AWS risorse e applicazioni che esegui 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 del tuo AWS account per Amazon GameLift 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

- [Monitora Amazon GameLift con Amazon CloudWatch](#)
- [Registrazione delle chiamate GameLift API Amazon con AWS CloudTrail](#)
- [Registrazione dei messaggi del server in Amazon GameLift](#)

Monitora Amazon GameLift con Amazon CloudWatch

Puoi monitorare Amazon GameLift utilizzando AmazonCloudWatch, un AWS servizio che raccoglie dati grezzi e li elabora in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per 15 mesi per fornire una panoramica storica delle prestazioni del tuo server di gioco che GameLift ospita su Amazon. 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 tabelle seguenti elencano le metriche e le dimensioni per AmazonGameLift. Tutte le metriche disponibili in CloudWatch sono disponibili anche nella GameLift console Amazon, che fornisce i dati sotto forma di grafici personalizzabili. Per accedere ai parametri di CloudWatch dei giochi, puoi utilizzare la AWS Management Console, l'AWS CLI o l'API CloudWatch.

Se una metrica non dispone di una posizione, utilizza la posizione della casa.

Dimensioni per le GameLift metriche Amazon

Amazon GameLift supporta il filtraggio delle metriche in base alle seguenti dimensioni.

Dimensione	Descrizione
Location	Filtra le metriche per una sede di distribuzione della flotta. Se una metrica non dispone di una posizione, utilizza la posizione della casa.
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 una flotta a un gruppo di metriche aggiungendo il nome del gruppo di metriche agli attributi della flotta (vedi UpdateFleetAttributes()). Questa dimensione può essere utilizzata con tutti i parametri di parchi istanze per istanze, processi server, sessioni di gioco e sessioni giocatore.

Dimensione	Descrizione
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 utilizzata a solo con i parametri per le configurazioni di abbinamenti.
ConfigurationName-RuleName	Filtra i parametri per l'intersezione di una configurazione di abbinamento e una regola di abbinamento. Questa dimensione viene utilizzata solo con parametri per regole di abbinamento.
InstanceType	Filtra i parametri per la designazione di un tipo di istanza EC2, come "c4.large". Questa dimensione viene utilizzata con parametri per istanze Spot.
OperatingSystem	Filtra i parametri per il sistema operativo di un'istanza. Questa dimensione viene utilizzata con i parametri per le istanze Spot.
GameServerGroup	Filtra le metriche di FleetIQ per un gruppo di server di gioco.

GameLiftMetriche Amazon per le flotte

Il namespace `AWS/GameLift` include i seguenti parametri relativi all'attività in un parco istanze o un gruppo di parchi istanze. Le flotte vengono utilizzate con una GameLift soluzione Amazon gestita. Il GameLift servizio Amazon invia le metriche a CloudWatch ogni minuto.

Istanze

Parametro	Descrizione
ActiveInstances	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
DesiredInstances	<p>Numero target di istanze attive che Amazon GameLift sta cercando di mantenere nel parco veicoli. 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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
IdleInstances	<p>Istanze attive che ospitano attualmente zero (0) sessioni di gioco. Questo parametro misura la capacità disponibile ma non utilizzata, e può essere utilizzato con la scalabilità automatica.</p>

Parametro	Descrizione
	<p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
MaxInstances	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
MinInstances	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
<code>PercentIdleInstances</code>	<p>Percentuale di tutte le istanze attive che sono inattive (calcolata come <code>IdleInstances / ActiveInstances</code>). Questo parametro può essere utilizzato per la scalabilità automatica.</p> <p>Unità: percentuale</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
<code>RecycledInstances</code>	<p>Numero di istanze spot che sono state riciclate e sostituite. Amazon GameLift ricicla le istanze spot che attualmente non ospitano sessioni di gioco e hanno un'alta probabilità di interruzione.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>
<code>InstanceInterruptions</code>	<p>Numero di istanze Spot che sono state interrotte.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
CPUUtilization	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in tutte le istanze attive in una flotta. La percentuale di tempo fisico della CPU utilizzata da Amazon EC2 per eseguire l'istanza, che include il tempo impiegato per eseguire sia il codice utente che il codice Amazon EC2. Gli strumenti del sistema operativo possono mostrare una percentuale diversa rispetto a quella CloudWatch dovuta a fattori quali la simulazione di dispositivi obsoleti, la configurazione di dispositivi non legacy, i carichi di lavoro con interruzioni elevate, la migrazione in tempo reale e l'aggiornamento in tempo reale.</p> <p>Unità: percentuale</p>
NetworkIn	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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.</p> <p>Unità: byte</p>
NetworkOut	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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.</p> <p>Unità: byte</p>

Parametro	Descrizione
DiskReadBytes	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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 determinare il volume dei dati che l'applicazione legge dal disco rigido dell'istanza. Puoi usarlo per determinare la velocità dell'applicazione.</p> <p>Unità: byte</p>
DiskWriteBytes	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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 determinare il volume dei dati che l'applicazione scrive sul disco rigido dell'istanza. Puoi usarlo per determinare la velocità dell'applicazione.</p> <p>Unità: byte</p>
DiskReadOps	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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 operazioni I/O medie al secondo (IOPS) per il periodo, dividi il numero di operazioni totali nel periodo per il numero di secondi in quel periodo.</p> <p>Unità: numero</p>

Parametro	Descrizione
DiskWriteOps	<p>Sistema metrico EC2. Per Amazon, GameLift questa metrica rappresenta le prestazioni hardware in 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 operazioni I/O medie al secondo (IOPS) per il periodo, dividi il numero di operazioni totali nel periodo per il numero di secondi in quel periodo.</p> <p>Unità: numero</p>

Processi server

Parametro	Descrizione
ActiveServerProcesses	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
HealthyServerProcesses	<p>Processi server attivi che risultano integri. Questo è un parametro utile per tenere traccia dello stato complessivo dei server di gioco del parco istanze.</p> <p>Unità: numero</p>


Parametro	Descrizione
	<p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
<p>PercentHealthyServerProcesses</p>	<p>Percentuale di tutti i processi server attivi che risultano integri (calcolata come <code>HealthyServerProcesses / ActiveServerProcesses</code>).</p> <p>Unità: percentuale</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
<p>ServerProcessAbnormalTerminations</p>	<p>Processi server che sono stati chiusi a causa di circostanze anomale dall'ultimo rapporto. Questa metrica include le terminazioni avviate dal servizio Amazon. GameLift Ciò si verifica quando un processo del server smette di rispondere, segnala costantemente i controlli di integrità non riusciti o non termina correttamente (ProcessEndingchiamando ()).</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
<code>ServerProcessActivations</code>	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>
<code>ServerProcessTerminations</code>	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>

Sessioni di gioco

Parametro	Descrizione
<code>ActivatingGameSessions</code>	<p>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</p>

Parametro	Descrizione
	<p>in corso) allo stato ACTIVE (Attivo). e può essere utilizzato con la scalabilità automatica.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
ActiveGameSessions	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
<code>AvailableGameSessions</code>	<p>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.</p> <div data-bbox="750 541 1507 999"><p> Note</p><p>Per le flotte che limitano le attivazioni simultanee delle sessioni di gioco, usa la metrica <code>ConcurrentActiveGameSessions</code>. Questa metrica rappresenta in modo più accurato il numero di nuove sessioni di gioco che possono essere avviate senza alcun tipo di ritardo.</p></div> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
<code>ConcurrentActivatableGameSessions</code>	<p>Processi server attivi e integri che non vengono attualmente utilizzati per ospitare una sessione di gioco e che possono avviare immediatamente una nuova sessione di gioco.</p> <p>Questa metrica differisce dalla <code>AvailableGameSessions</code> seguente: non tiene conto dei 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 RuntimeConfiguration opzionale della flotta <code>MaxConcurrentGameSessionActivations</code>). Per le flotte che non limitano le attivazioni delle sessioni di gioco, questa metrica è identica a <code>AvailableGameSessions</code>.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: posizione</p>
<code>PercentAvailableGameSessions</code>	<p>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 $\text{AvailableGameSessions} / [\text{ActiveGameSessions} + \text{AvailableGameSessions} + \text{unhealthy server processes}]$), e può essere utilizzato con la scalabilità automatica.</p> <p>Unità: percentuale</p> <p>CloudWatchStatistiche pertinenti: media</p> <p>Dimensioni: posizione</p>

Parametro	Descrizione
GameSessionInterruptions	<p>Il numero di sessioni di gioco su istanze Spot che sono state interrotte.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p> <p>Dimensioni: posizione</p>

Sessioni giocatore

Parametro	Descrizione
CurrentPlayerSessions	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p>
PlayerSessionActivations	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma, media, minima, massima</p>

GameLiftMetriche Amazon per le 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 gestita. Il GameLift servizio Amazon invia le metriche a CloudWatch ogni minuto.

Parametro	Descrizione
AverageWaitTime	<p>Il tempo medio di attesa prima che le richieste di posizionamento delle sessioni di gioco in coda con stato IN SOSPESO vengano soddisfatte.</p> <p>Unità: secondi</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p> <p>Dimensioni: posizione</p>
FirstChoiceNotViable	<p>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 nella coda oppure, quando una richiesta di piazzamento include i dati di latenza dei giocatori, è la prima flotta scelta dalla priorità di FleetIQ. Se non ci sono parchi Spot realizzabili, potrebbe essere selezionato qualsiasi parco istanze in quella regione.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
FirstChoiceOutOfCapacity	<p>Sessioni di gioco correttamente piazzate ma NON nel parco istanze di prima scelta, perché tale parco istanze non aveva risorse disponibili. La flotta</p>

Parametro	Descrizione
	<p>di prima scelta è la prima flotta elencata in coda oppure, quando una richiesta di piazzamento include dati sulla latenza dei giocatori, è la prima flotta scelta in base alla priorità definita da FleetIQ.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
LowestLatencyPlacement	<p>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.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
LowestPricePlacement	<p>Sessioni di gioco inserite con successo in una flotta con il prezzo più basso possibile in coda per la regione scelta. 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 posizionamento.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>

Parametro	Descrizione
<code>Placement <region name></code>	<p>Le sessioni di gioco correttamente piazzate nei parchi istanze posizionati nella regione specificata. Questo parametro scompone il parametro <code>PlacementsSucceeded</code> per regione.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
<code>PlacementsCanceled</code>	<p>Le richieste di posizionamento delle sessioni di gioco che sono state annullate prima del timeout dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
<code>PlacementsFailed</code>	<p>Le richieste di posizionamento delle sessioni di gioco non andate a buon fine per qualsiasi motivo dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
<code>PlacementsStarted</code>	<p>Le nuove richieste di posizionamento delle sessioni di gioco aggiunte alla coda dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>

Parametro	Descrizione
PlacementsSucceeded	<p>Le richieste di posizionamento delle sessioni di gioco che hanno avuto come esito una nuova sessione di gioco dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
PlacementsTimedOut	<p>Le richieste di posizionamento delle sessioni di gioco che hanno raggiunto il limite di timeout della coda senza venire soddisfatte dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
QueueDepth	<p>Il numero di richieste di posizionamento delle sessioni di gioco in coda con stato IN SOSPESO.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p> <p>Dimensioni: posizione</p>

GameLiftMetriche Amazon per il matchmaking

Il Amazon GameLift namespace include metriche sull'FlexMatchattività per le configurazioni di matchmaking e le regole di matchmaking. FlexMatchil matchmaking viene utilizzato con una GameLift soluzione Amazon gestita. Il GameLift servizio Amazon invia le metriche a CloudWatch ogni minuto.

Per ulteriori informazioni sulla sequenza delle attività di matchmaking, consulta [How Amazon GameLift FlexMatch works](#).

Configurazioni di abbinamento

Parametro	Descrizione
CurrentTickets	<p>Le richieste di abbinamento attualmente in fase di elaborazione o in attesa di essere elaborate.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima, somma</p>
MatchAcceptancesTimedOut	<p>Per le configurazioni di abbinamento che richiedono o accettazione, i potenziali abbinamenti scaduti durante l'accettazione dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
MatchesAccepted	<p>Per le configurazioni di abbinamento che richiedono accettazione, i potenziali abbinamenti che sono stati accettati dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
MatchesCreated	<p>I potenziali abbinamenti che sono stati creati dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
MatchesPlaced	<p>Gli abbinamenti che sono stati posizionati correttamente in una sessione di gioco dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>

Parametro	Descrizione
MatchesRejected	<p>Per le configurazioni di abbinamento che richiedono accettazione, i potenziali abbinamenti che sono stati rifiutati da almeno un giocatore dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
PlayersStarted	<p>I giocatori in ticket di abbinamento che sono stati aggiunti dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
TicketsFailed	<p>Le richieste di abbinamento che hanno prodotto un errore dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
TicketsStarted	<p>Le nuove richieste di abbinamento che sono state create dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>
TicketsTimedOut	<p>Le richieste di abbinamento che hanno raggiunto il limite di timeout dall'ultimo rapporto.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p>

Parametro	Descrizione
<code>TimeToMatch</code>	<p>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.</p> <p>Unità: secondi</p> <p>CloudWatchStatistiche pertinenti: campioni di dati, media, minima, massima</p>
<code>TimeToTicketCancel</code>	<p>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.</p> <p>Unità: secondi</p> <p>CloudWatchStatistiche pertinenti: campioni di dati, media, minima, massima</p>
<code>TimeToTicketSuccess</code>	<p>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.</p> <p>Unità: secondi</p> <p>CloudWatchStatistiche pertinenti: campioni di dati, media, minima, massima</p>

Regole di abbinamento

Parametro	Descrizione
<code>RuleEvaluationsPassed</code>	<p>Le valutazioni delle regole durante il processo di abbinamento superato dall'ultimo rapporto. Questo parametro si limita alle prime 50 regole.</p>

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 CloudWatchStatistiche pertinenti: somma

GameLiftMetriche Amazon per FleetIQ

Il Amazon GameLift namespace include metriche per il gruppo di server di gioco FleetIQ e l'attività dei server di gioco come parte di una soluzione autonoma FleetIQ per l'hosting di giochi. Il GameLift servizio Amazon invia le metriche a CloudWatch ogni minuto. Consulta anche [Monitoraggio dei gruppi e delle istanze di Auto Scaling utilizzando Amazon CloudWatch nella Guida per l'utente](#) di Amazon EC2 Auto Scaling.

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 CloudWatchStatistiche pertinenti: somma 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 CloudWatchStatistiche pertinenti: somma Dimensioni: GameServerGroup
<code>DrainingAvailableGameServers</code>	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 CloudWatchStatistiche pertinenti: somma Dimensioni: GameServerGroup
<code>DrainingUtilizedGameServers</code>	Server di gioco su istanze programmate per la terminazione che attualmente supportano la modalità di gioco. Unità: numero CloudWatchStatistiche pertinenti: somma Dimensioni: GameServerGroup

Parametro	Descrizione
<code>PercentUtilizedGameServers</code>	<p>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 aggiungere e rimuovere dinamicamente istanze in base alla domanda dei giocatori.</p> <p>Unità: percentuale</p> <p>CloudWatchStatistiche pertinenti: media, minima, massima</p> <p>Dimensioni: <code>GameServerGroup</code></p>
<code>GameServerInterruptions</code>	<p>Server di gioco su istanze Spot che sono state interrotte a causa della disponibilità Spot limitata.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p> <p>Dimensioni: <code>GameServerGroup</code>, <code>InstanceType</code></p>
<code>InstanceInterruptions</code>	<p>Istanze Spot che sono state interrotte a causa della disponibilità limitata.</p> <p>Unità: numero</p> <p>CloudWatchStatistiche pertinenti: somma</p> <p>Dimensioni: <code>GameServerGroup</code>, <code>InstanceType</code></p>

Registrazione delle chiamate GameLift API Amazon con AWS CloudTrail

Amazon GameLift è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, da un ruolo o da un AWS servizio in AmazonGameLift. CloudTrail acquisisce tutte le chiamate API per Amazon GameLift come eventi. Le chiamate acquisite includono chiamate dalla GameLift console Amazon e chiamate in codice alle operazioni dell'GameLift API Amazon. Se crei un percorso, puoi abilitare la distribuzione continua di CloudTrail eventi su un bucket Amazon S3, inclusi gli eventi per Amazon GameLift. Se non configuri un trail, è comunque possibile visualizzare gli eventi più recenti nella console di CloudTrail in Event history (Cronologia eventi). Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad AmazonGameLift, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni su CloudTrail, consulta la [Guida per l'utente di AWS CloudTrail](#).

GameLift Informazioni su Amazon in CloudTrail

CloudTrail è abilitato sull'Account AWS al momento della sua creazione. Quando si verifica un'attività su AmazonGameLift, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nell'Account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi mediante la cronologia eventi di CloudTrail](#).

Per una registrazione continua dei tuoi eventi Account AWS, inclusi quelli per AmazonGameLift, crea un percorso. Un percorso consente di CloudTrail inviare file di registro a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il percorso registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri servizi AWS per analizzare con maggiore dettaglio e usare i dati raccolti nei log CloudTrail. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [Servizi e integrazioni CloudTrail supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di log CloudTrail da più regioni](#) e [Ricezione di file di log CloudTrail da più account](#)

Tutte le GameLift azioni di Amazon vengono registrate CloudTrail e documentate nell'[Amazon GameLift API Reference](#). 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 servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Comprendere le voci dei file di GameLift registro di Amazon

Un percorso è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato. I file di log di CloudTrail contengono una o più voci di log. Un evento rappresenta una singola richiesta da un'fonte e include informazioni sul operazione richiesta, data e ora dell'operazione, parametri richiesti e così via. I file di log di CloudTrail non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di log di CloudTrail che illustra le operazioni `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 in Amazon GameLift

Puoi acquisire messaggi server personalizzati dai tuoi GameLift server Amazon in file di registro. La modalità di configurazione della registrazione dipende dal fatto che si utilizzino server personalizzati o server in tempo reale (vedere le sottosezioni appropriate in questo capitolo).

Argomenti

- [Messaggi del server di registrazione \(server personalizzati\)](#)
- [Messaggi del server di registrazione \(Realtime Servers\)](#)

Messaggi del server di registrazione (server personalizzati)

Puoi acquisire messaggi server personalizzati dai tuoi server Amazon GameLift personalizzati in file di registro. Per ulteriori informazioni sulla registrazione per i server in tempo reale, consulta.

[Messaggi del server di registrazione \(Realtime Servers\)](#)

Important

Esiste un limite alla dimensione di un file di registro per sessione di gioco (consulta gli [GameLift endpoint e le quote di Amazon](#) nel Riferimenti generali di AWS). Al termine di una sessione di gioco, Amazon GameLift carica i log del server su Amazon Simple Storage Service (Amazon S3). Amazon non GameLift 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 GameLift personalizzati Amazon, scrivi il tuo codice per eseguire la registrazione, che configuri come parte della configurazione del processo del server. Amazon GameLift utilizza la tua configurazione di registrazione per identificare i file da 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 [LogParameters](#) del 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'[ProcessParameters](#) oggetto quando chiami [ProcessReady\(\)](#).

```
Aws::GameLift::GenericOutcome outcome =
  Aws::GameLift::Server::ProcessReady(processReadyParameter);
```

Per un esempio più completo, vedi [ProcessReady\(\)](#).

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 [ProcessParameters](#) oggetto. [LogParameters](#)

```
var processReadyParameter = new ProcessParameters(
  this.OnGameSession,
  this.OnProcessTerminate,
  this.OnHealthCheck,
  this.OnGameSessionUpdate,
  port,
  new LogParameters(logPaths));
```

3. Fornisci l'[ProcessParameters](#) oggetto quando chiami [ProcessReady\(\)](#).

```
var processReadyOutcome =
```



```
GameLiftServerAPI.ProcessReady(processReadyParameter);
```

Per un esempio più completo, vedi [ProcessReady\(\)](#).

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, Amazon archivia GameLift 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 (Realtime Servers)

Puoi acquisire messaggi server personalizzati dai tuoi server Realtime in file di registro. Per ulteriori informazioni sulla registrazione per server personalizzati, consulta [Messaggi 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, i server Realtime emettono messaggi di sistema utilizzando gli stessi tipi di messaggi e scrivono negli stessi file di registro. Puoi regolare il livello di registrazione del tuo parco veicoli per ridurre la quantità di messaggi di registrazione generati dai server (vedi). [Regolazione del livello di registrazione](#)

⚠ Important

Esiste un limite alla dimensione di un file di registro per sessione di gioco (vedi [GameLift Endpoints e quote Amazon](#) nella Riferimenti generali di AWS). Al termine di una sessione di gioco, Amazon GameLift carica i log del server su Amazon Simple Storage Service (Amazon S3). Amazon non GameLift 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 i tuoi server in tempo reale](#). 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. [Esempio di script Realtime Servers](#)

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, Amazon archivia GameLift 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. Per i server in tempo reale, 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 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 una flotta di server in tempo reale (console)

Segui le istruzioni riportate [Crea una flotta GameLift gestita da Amazon](#) 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 di server in tempo reale () AWS CLI

Segui le istruzioni riportate [Crea una flotta GameLift gestita da Amazon](#) 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 una flotta di server Realtime in esecuzione (console)

Segui le istruzioni riportate [Aggiornare la configurazione di una flotta](#) per aggiornare la tua flotta utilizzando la GameLift console Amazon, 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 di server Realtime in esecuzione () AWS CLI

Segui le istruzioni riportate [Aggiornare la configurazione di una flotta](#) 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

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

La sicurezza del cloud è la massima priorità. AWS 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 maggiori informazioni sui programmi di conformità applicabili ad Amazon GameLift, consulta la sezione [AWS Servizi rientranti nell'ambito del programma di conformità](#), .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei responsabile anche di altri fattori, tra cui la sensibilità dei tuoi dati, i requisiti aziendali AWS e le normative applicabili.

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon GameLift. I seguenti argomenti mostrano come configurare Amazon per GameLift soddisfare i tuoi obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue GameLift risorse Amazon.

Argomenti

- [Protezione dei dati in Amazon GameLift](#)
- [Gestione delle identità e degli accessi per Amazon GameLift](#)
- [Registrazione e monitoraggio con Amazon GameLift](#)
- [Convalida della conformità per Amazon GameLift](#)
- [Resilienza in Amazon GameLift](#)
- [Sicurezza dell'infrastruttura in Amazon GameLift](#)
- [Analisi della configurazione e delle vulnerabilità in Amazon GameLift](#)
- [Best practice di sicurezza per Amazon GameLift](#)

Protezione dei dati in Amazon GameLift

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

Il modello di [responsabilità AWS condivisa Modello](#) di si applica alla protezione dei dati in Amazon GameLift. 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.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- 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 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente 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 con Amazon GameLift o altri utenti Servizi AWS utilizzando la console, l'API o AWS gli SDK. AWS CLI 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 GameLift specifici di Amazon vengono gestiti come segue:

- Le build dei server di gioco e gli script che carichi su Amazon GameLift 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, usa la GameLift console Amazon 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 registro scaricandoli tramite un collegamento nella GameLift console Amazon o tramite chiamate all'API del servizio.
- I dati relativi a metriche ed eventi sono archiviati in Amazon GameLift e sono accessibili tramite la GameLift console Amazon o tramite 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 archiviati in Amazon. GameLift 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 fornisci ID giocatore personalizzati nelle tue richieste, è previsto che questi valori siano UUID anonimizzati 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 a riposo dei dati GameLift specifici di Amazon 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 GameLift in Amazon in un formato crittografato.

Crittografia in transito

Le connessioni alle GameLift API di Amazon 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 su GameLift risorse Amazon, la comunicazione non coinvolge il GameLift servizio Amazon. 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.
- Per i server Realtime con la generazione di certificati TLS abilitata, il traffico tra il client di gioco e i server Realtime che utilizza Realtime Client SDK viene crittografato durante il trasferimento. Il traffico TCP viene crittografato con TLS 1.2 e il traffico UDP viene crittografato con DTLS 1.2.

Riservatezza del traffico Internet

Puoi accedere in remoto alle tue GameLift istanze Amazon 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 FleetIQ, 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 assistenza sulla connessione a un' GameLift istanza Amazon gestita, consulta [Connessione remota alle istanze GameLift della flotta Amazon](#).

Gestione delle identità e degli accessi per Amazon GameLift

AWS Identity and Access Management (IAM) è un Servizio AWS che consente agli amministratori di controllare in modo sicuro l'accesso alle risorse AWS. Gli amministratori IAM controllano chi può

essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse Amazon. GameLift IAM è un Servizio AWS il cui uso non comporta costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come GameLift funziona Amazon con IAM](#)
- [Esempi di policy basate sull'identità per Amazon GameLift](#)
- [Risoluzione dei problemi relativi all' GameLift identità e all'accesso ad Amazon](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon GameLift.

Utente del servizio: se utilizzi il GameLift servizio Amazon per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più GameLift funzionalità di Amazon per svolgere 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 riesci ad accedere a una funzionalità di Amazon GameLift, consulta [Risoluzione dei problemi relativi all' GameLift identità e all'accesso ad Amazon](#).

Amministratore del servizio: se sei responsabile delle GameLift risorse Amazon della tua azienda, probabilmente hai pieno accesso ad Amazon GameLift. È tuo compito determinare a quali GameLift funzionalità e risorse di Amazon devono accedere gli utenti del servizio. 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 ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon GameLift, consulta [Come GameLift funziona Amazon con IAM](#).

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso ad Amazon GameLift. Per visualizzare esempi di policy GameLift basate sull'identità di Amazon che puoi utilizzare in IAM, consulta. [Esempi di policy basate sull'identità per Amazon GameLift](#)

Autenticazione con identità

L'autenticazione è la procedura di accesso ad AWS con le credenziali di identità. Devi essere autenticato (connesso a AWS) come utente root Utente root dell'account AWS, come utente IAM o assumere un ruolo IAM.

Puoi accedere ad AWS come identità federata utilizzando le credenziali fornite attraverso un'origine di identità. Gli utenti AWS IAM Identity Center (Centro identità IAM), l'autenticazione Single Sign-On (SSO) dell'azienda e le 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. Se accedi ad AWS tramite la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere alla AWS Management Console o al portale di accesso AWS. Per ulteriori informazioni sull'accesso ad AWS, consulta la sezione [Come accedere al tuo Account AWS](#) nella Guida per l'utente di Accedi ad AWS.

Se accedi ad AWS in modo programmatico, AWS fornisce un Software Development Kit (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le richieste utilizzando le tue credenziali. Se non utilizzi gli strumenti AWS, devi firmare le richieste personalmente. Per ulteriori informazioni sulla firma delle richieste, consulta [Firma delle richieste AWS](#) nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. AWS consiglia ad esempio di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza dell'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 \(MFA\) in AWS](#) nella Guida per l'utente di IAM.

Utente root di un Account AWS

Quando crei un Account AWS, inizi con una singola identità di accesso che ha accesso completo a tutti i Servizi AWS e le risorse nell'account. Tale identità è detta utente root Account AWS ed è possibile accedervi 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 utilizzarle 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 di IAM.

Identità federata

Come best practice, richiedi agli utenti umani, compresi quelli che richiedono l'accesso di amministratore, di utilizzare la federazione con un provider di identità per accedere a Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory degli utenti aziendali, un provider di identità Web, AWS Directory Service, la directory Identity Center o qualsiasi utente che accede ai Servizi AWS utilizzando le credenziali fornite tramite un'origine di identità. Quando le identità federate accedono agli Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. È possibile creare utenti e gruppi in IAM Identity Center oppure connettersi e sincronizzarsi con un gruppo di utenti e gruppi nell'origine di identità per utilizzarli in tutte le applicazioni e gli Account AWS. Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center.

Utenti e gruppi IAM

Un [utente IAM](#) è una identità all'interno del tuo 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, per 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 di 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, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale 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 [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità all'interno di un Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. È possibile assumere temporaneamente un ruolo IAM nella AWS Management Console mediante lo [scambio di ruoli](#). È possibile assumere un ruolo chiamando un'azione AWS CLI o API AWS oppure utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di 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 [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di 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 ulteriori 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, per alcuni dei Servizi AWS, è possibile collegare 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 [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso multi-servizio:** alcuni Servizi AWS utilizzano funzionalità in altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale 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.
 - **Inoltro delle sessioni di accesso (FAS):** quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, tale utente o ruolo viene considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio

AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) assunto da un servizio per eseguire operazioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di 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 ai servizi sono visualizzati nell'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:** è possibile utilizzare un ruolo IAM per gestire credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 che eseguono richieste di AWS CLI o dell'API AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2, affinché sia disponibile per tutte le relative applicazioni, puoi creare un profilo dell'istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Per controllare l'accesso a AWS è possibile creare policy e collegarle a identità o risorse AWS. Una policy è un oggetto in AWS che, quando associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste policy quando un principale IAM (utente, utente root o sessione ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle policy viene archiviata in AWS sotto forma di 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 di IAM.

Gli amministratori possono utilizzare le policy AWSJSON per specificare l'accesso ai diversi elementi. In altre parole, quale principale può eseguire azioni 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 azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può 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'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dalla AWS Management Console, la AWS CLI o l'API AWS.

Policy basate su 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 azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le policy gestite sono policy autonome che possono essere collegate a più utenti, gruppi e ruoli in Account AWS. Le policy gestite includono le policy gestite da AWS e le policy gestite dal cliente. 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 di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su 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 utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni 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 è possibile utilizzare le policy gestite da AWS da IAM in una policy basata su risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3, AWS WAF e Amazon VPC sono esempi di servizi che supportano le ACL. Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta altri tipi di policy 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 funzione 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 di IAM.
- **Policy di controllo dei servizi (SCP):** le SCP sono policy JSON che specificano il numero massimo di autorizzazioni per un'organizzazione o unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata degli Account AWS multipli di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. La SCP limita le autorizzazioni per le entità negli account membri, compreso ogni Utente root dell'account AWS. Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.
- **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 di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per informazioni su come AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella Guida per l'utente di IAM.

Come GameLift funziona Amazon con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon GameLift, scopri quali funzionalità IAM sono disponibili per l'uso con Amazon GameLift.

Funzionalità IAM che puoi utilizzare con Amazon GameLift

Funzionalità IAM	GameLift Assistenza Amazon
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
Liste di controllo degli accessi (ACL)	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 e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWSi servizi che funzionano con IAM nella IAM User Guide](#).

Politiche basate sull'identità per Amazon GameLift

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 azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di 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 policy basate sull'identità per Amazon GameLift

Per visualizzare esempi di politiche GameLift basate sull'identità di Amazon, consulta [Esempi di policy basate sull'identità per Amazon GameLift](#)

Politiche basate sulle risorse all'interno di Amazon GameLift

Supporta le policy basate su risorse No

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su 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 utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni 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 l'entità principale e la risorsa si trovano in diversi Account AWS, un amministratore IAM nell'account attendibile deve concedere all'entità principale (utente o ruolo) anche l'autorizzazione per 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 [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

Azioni politiche per Amazon GameLift

Supporta le azioni di policy

Sì

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni di policy hanno spesso lo stesso nome dell'operazione API AWS. Ci sono alcune eccezioni, ad esempio le azioni 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 delle GameLift azioni Amazon, consulta [Azioni definite da Amazon GameLift](#) nel Service Authorization Reference.

Le azioni politiche in Amazon GameLift utilizzano 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 operazioni che iniziano con la parola `Describe`, includi la seguente operazione:

```
"Action": "gamelift:Describe*"
```

Per visualizzare esempi di politiche GameLift basate sull'identità di Amazon, consulta [Esempi di policy basate sull'identità per Amazon GameLift](#)

Risorse relative alle policy per Amazon GameLift

Supporta le risorse di policy

Si

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, 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'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni 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 dei tipi di GameLift risorse Amazon e dei relativi ARN, consulta [Resources defined by Amazon GameLift](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta [Azioni definite da Amazon](#). GameLift

Alcune GameLift risorse Amazon hanno valori ARN, che consentono di gestire l'accesso alle risorse utilizzando le policy IAM. La risorsa GameLift della flotta Amazon ha un ARN con la seguente sintassi:

```
arn:${Partition}:gamelift:${Region}:${Account}:fleet/${FleetId}
```

Per ulteriori informazioni sul formato degli ARN, consulta [Amazon Resource Names \(ARN\)](#) nella 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 politiche GameLift basate sull'identità di Amazon, consulta. [Esempi di policy basate sull'identità per Amazon GameLift](#)

Chiavi relative alle condizioni delle politiche per Amazon GameLift

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì
---	----

Gli amministratori possono utilizzare le policy JSON AWS per specificare gli accessi ai diversi elementi. Cioè, quale principale può eseguire azioni 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. Puoi 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 specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione OR logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi 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 per il servizio. Per visualizzare tutte le chiavi di condizione globali di AWS, consulta [Chiavi di contesto delle condizioni globali di AWS](#) nella Guida per l'utente di IAM.

Per un elenco delle chiavi di GameLift condizione di Amazon, consulta [Condition keys for Amazon GameLift](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Azioni definite da Amazon GameLift](#).

Per visualizzare esempi di politiche GameLift basate sull'identità di Amazon, consulta [Esempi di policy basate sull'identità per Amazon GameLift](#)

ACL su Amazon GameLift

Supporta le ACL

No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Amazon GameLift

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, tali attributi sono denominati tag. È possibile collegare dei tag alle entità IAM (utenti o ruoli) e a numerose risorse AWS. 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 [Che cos'è ABAC?](#) nella Guida per l'utente di 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. [Visualizza le GameLift flotte Amazon in base ai tag](#)

Utilizzo di credenziali temporanee con Amazon GameLift

Supporta le credenziali temporanee	Sì
------------------------------------	----

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, inclusi i Servizi AWS che funzionano con le credenziali temporanee, consulta [Servizi AWS supportati da IAM](#) nella Guida per l'utente IAM.

Le credenziali temporanee sono utilizzate se si accede alla AWS Management Console utilizzando qualsiasi metodo che non sia la combinazione di nome utente e password. Ad esempio, quando accedi ad AWS utilizzando il collegamento Single Sign-On (SSO) della tua azienda, tale processo crea in automatico 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 [Cambio di un ruolo \(console\)](#) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando la AWS CLI o l'API AWS. È quindi possibile utilizzare tali credenziali temporanee per accedere ad AWS. AWS consiglia di generare le credenziali temporanee dinamicamente anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

Autorizzazioni principali multiservizio per Amazon GameLift

Supporta sessioni di accesso diretto (FAS)	Sì
--	----

Quando si utilizza un utente o un ruolo IAM per eseguire operazioni in AWS, si viene considerati un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'azione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che effettua la chiamata a un Servizio AWS, combinate con il Servizio AWS richiedente, per effettuare richieste a servizi a valle. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che necessita di interazioni con altri Servizi AWS o risorse per essere portata a termine. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

Ruoli di servizio per Amazon GameLift

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 [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la GameLift funzionalità di Amazon. Modifica i ruoli di servizio solo quando Amazon GameLift fornisce indicazioni in tal senso.

Consenti ai tuoi server GameLift di gioco ospitati su Amazon di accedere ad altre AWS risorse, come una AWS Lambda funzione o un database Amazon DynamoDB. Poiché i server di gioco sono ospitati su flotte GameLift gestite da Amazon, è necessario un ruolo di servizio che offra ad Amazon un accesso GameLift limitato alle altre AWS risorse. Per ulteriori informazioni, consulta [Comunica con altre AWS risorse delle tue flotte](#).

Ruoli collegati ai servizi per Amazon GameLift

Supporta i ruoli collegati ai servizi	No
---------------------------------------	----

Un ruolo collegato ai servizi è un tipo di ruolo di servizio che è collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'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 [AWSservizi 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

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare GameLift risorse Amazon. Inoltre, non sono in grado di eseguire attività utilizzando la AWS Management Console, l'AWS Command Line Interface (AWS CLI) o l'API AWS. Per concedere agli utenti l'autorizzazione per 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](#) nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da Amazon GameLift, incluso il formato degli ARN per ciascun tipo di risorsa, consulta [Azioni, risorse e chiavi di condizione per Amazon GameLift](#) nel Service Authorization Reference.

Argomenti

- [Best practice per le policy](#)
- [Utilizzo della GameLift console Amazon](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Consentire l'accesso ai giocatori per le sessioni di gioco](#)
- [Consenti l'accesso a una GameLift coda Amazon](#)
- [Visualizza le GameLift flotte Amazon in base ai 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 GameLift risorse Amazon 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:

- Nozioni di base sulle policy gestite da AWS e passaggio alle autorizzazioni con privilegio minimo: per le informazioni di base su come concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite da AWS che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente di AWS specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi 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 di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi 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 inoltre utilizzare le condizioni per concedere l'accesso alle operazioni di servizio, ma solo 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 di 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 IAM Access Analyzer](#) nella Guida per l'utente di IAM.
- Richiesta dell'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o utenti root nel tuo Account AWS, attiva 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 [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della GameLift console Amazon

Per accedere alla GameLift console Amazon, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle GameLift risorse Amazon 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 la GameLift console Amazon, aggiungi le autorizzazioni a utenti e gruppi con la sintassi riportata negli esempi seguenti e versioni successive. [Esempi di autorizzazioni di amministratore](#) Per ulteriori informazioni, consulta [Gestisci le autorizzazioni degli utenti per Amazon GameLift](#).

Gli utenti che collaborano con Amazon GameLift tramite le AWS CLI nostre operazioni AWS 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 clienti di gioco, richiede l'accesso per richiedere sessioni di gioco, inserire giocatori nei giochi e altre attività.

Per informazioni sulle autorizzazioni necessarie per utilizzare tutte le funzionalità della GameLift console Amazon, consulta la sintassi delle autorizzazioni per gli amministratori in [Esempi di autorizzazioni di amministratore](#)

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 allegate alla relativa identità utente. La policy include le autorizzazioni per completare questa azione sulla console o a livello di programmazione utilizzando la AWS CLI o l'API AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
```

```

        "iam:ListGroupsWithUser",
        "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 degli utenti dei giocatori](#)

Consenti l'accesso a una GameLift coda Amazon

L'esempio seguente fornisce a un utente l'accesso a una specifica GameLift coda Amazon.

Questa politica concede all'utente le autorizzazioni per aggiungere, aggiornare ed eliminare le destinazioni delle code 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"
  }
]
}

```

Visualizza le GameLift flotte Amazon in base ai tag

Puoi utilizzare le condizioni della tua policy basata sull'identità per controllare l'accesso alle GameLift risorse Amazon in base ai tag. Questo esempio mostra come creare una policy 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 tuo server di gioco con Amazon GameLift, carica i file di build su Amazon S3. Per consentire GameLift ad Amazon di 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 dei file di GameLift gioco Amazon, consulta [Carica un server personalizzato su Amazon GameLift](#).

Risoluzione dei problemi relativi all' GameLift identità e all'accesso ad Amazon

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon GameLift e AWS Identity and Access Management (IAM).

Argomenti

- [Non sono autorizzato a eseguire un'azione in Amazon GameLift](#)

- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne Account AWS a me di accedere alle mie GameLift risorse Amazon](#)

Non sono autorizzato a eseguire un'azione in Amazon GameLift

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 trasferire un ruolo ad Amazon GameLift.

Alcuni Servizi AWS consentono di trasmettere un ruolo esistente a tale servizio, invece di creare un nuovo ruolo di servizio o un ruolo collegato ai servizi. 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. 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`.

Per ulteriore assistenza con l'accesso, contatta l'amministratore AWS. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne Account AWS a me di accedere alle mie GameLift risorse Amazon

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 servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon GameLift supporta queste funzionalità, consulta [Come GameLift funziona Amazon con IAM](#).
- Per informazioni su come garantire l'accesso alle risorse negli Account AWS che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS in tuo possesso](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso alle risorse ad Account AWS di terze parti, consulta [Fornire l'accesso agli 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 di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

Registrazione e monitoraggio con Amazon GameLift

Il monitoraggio è una parte importante del mantenimento dell'affidabilità, della disponibilità e delle prestazioni di Amazon GameLift e delle tue AWS soluzioni. È necessario raccogliere i dati sul monitoraggio da tutte le parti della soluzione AWS per consentire un debug più facile di eventuali guasti in più punti.

AWS e Amazon GameLift forniscono diversi strumenti per monitorare le risorse di hosting dei giochi e rispondere a potenziali incidenti.

CloudWatch Allarmi Amazon

Utilizzando gli CloudWatch allarmi Amazon, controlli una singola metrica in un periodo di tempo specificato. Se il parametro supera una determinata soglia, viene inviata una notifica a un argomento Amazon SNS o alla policy AWS Auto Scaling. 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 con Amazon CloudWatch](#).

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, da un ruolo o da un AWS servizio in Amazon GameLift. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon GameLift, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli. Per ulteriori informazioni, consulta [Registrazione delle chiamate GameLift API Amazon con AWS CloudTrail](#).

Convalida della conformità per Amazon GameLift

Amazon GameLift 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:

- [Guide introduttive su sicurezza e conformità](#): queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla AWS sicurezza e la conformità.

- [Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services](#): questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee all'HIPAA.

Note

Non tutti i Servizi AWS sono idonei all'HIPAA. Per ulteriori informazioni, consulta la sezione [Riferimenti sui servizi conformi ai requisiti HIPAA](#).

- [AWS Risorse 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)).
- [Evaluating Resources with Rules](#) nella AWS Config Developer Guide: 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

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

L'infrastruttura AWS globale è costruita attorno a 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 offre le seguenti funzionalità per aiutarti a supportare le tue esigenze di resilienza dei dati:

- **Code multiregionali:** le code delle sessioni di GameLift gioco di Amazon 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 [Progetta una coda per le sessioni di gioco](#).
- **Scalabilità automatica della capacità:** mantieni l'integrità e la disponibilità delle tue risorse di hosting utilizzando gli strumenti di GameLift scalabilità di Amazon. 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 GameLift hosting di Amazon](#).
- **Distribuzione tra istanze:** Amazon GameLift 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:** le build e gli script dei server di gioco caricati su Amazon GameLift vengono 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

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

In quanto servizio gestito, Amazon GameLift è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper [Amazon Web Services: panoramica dei processi di sicurezza](#).

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon GameLift tramite 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 GameLift servizio Amazon colloca tutte le flotte nei cloud privati virtuali (VPC) di Amazon in modo che ogni flotta esista in un'area logicamente isolata nel cloud. AWS Puoi utilizzare GameLift le policy di Amazon per controllare l'accesso da endpoint VPC specifici o VPC specifici. In effetti, questo isola l'accesso alla rete a una determinata GameLift risorsa Amazon solo dal VPC specifico all'interno AWS della 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

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

Configurazione e controllo IT sono una responsabilità condivisa tra AWS e te, il nostro cliente. [Per ulteriori informazioni, consulta il modello di responsabilità 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:

- I clienti sono responsabili della gestione del software distribuito GameLift sulle istanze Amazon 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, 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 l'aggiornamento regolare dei propri giochi con le versioni SDK più recenti, tra cui AWS SDK, Amazon GameLift Server SDK e Amazon GameLift Client SDK per server in tempo reale.

Best practice di sicurezza per Amazon GameLift

Se utilizzi Amazon GameLift FleetIQ come funzionalità autonoma con Amazon EC2, consulta la sezione Sicurezza [in Amazon EC2 nella Guida per l'utente di Amazon EC2](#).

Amazon GameLift offre una serie di funzionalità di sicurezza da prendere in considerazione durante lo sviluppo e l'implementazione delle proprie politiche 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 [UpdateFleetPortSettings](#) per 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 rendere GameLift più sicuro l'utilizzo di Amazon, consulta il [pilastro AWS Well-Architected Tool Sicurezza](#).

Guide GameLift di riferimento Amazon

Questa sezione contiene la documentazione di riferimento per l'utilizzo di AmazonGameLift.

Argomenti

- [Riferimento alle API dei GameLift servizi Amazon \(AWSSDK\)](#)
- [Riferimento Amazon GameLift Realtime Servers](#)
- [Riferimento SDK per GameLift server Amazon](#)
- [Eventi di collocamento delle sessioni di gioco](#)

Riferimento alle API dei GameLift servizi Amazon (AWSSDK)

Questo argomento fornisce un elenco basato sulle attività di operazioni API da utilizzare con le soluzioni di hosting GameLift gestito di Amazon, incluso l'hosting per server di gioco personalizzati e server in tempo reale. Queste operazioni sono incluse nell'AWSSDK nel namespace `aws.gamelift`. [Scarica l'AWSSDK](#) o [visualizza la documentazione di riferimento dell'GameLiftAPI Amazon](#).

L'API include due set di operazioni per l'hosting di giochi gestito:

- [Configura e gestisci le risorse GameLift di hosting Amazon](#)
- [Inizia sessioni di gioco e unisciti ai giocatori](#)

L'API Amazon GameLift Service contiene anche operazioni da utilizzare con altri GameLift strumenti e soluzioni Amazon. Per un elenco delle API FleetIQ, consulta le azioni dell'API [FleetIQ](#). Per un elenco di FlexMatch API per il matchmaking, vedi Azioni [FlexMatchAPI](#).

Configura e gestisci le risorse GameLift di hosting Amazon

Esegui queste operazioni per configurare le risorse di hosting per i tuoi server di gioco, scalare la capacità per soddisfare la domanda dei giocatori, accedere alle metriche di prestazioni e utilizzo e altro ancora. Queste operazioni API vengono utilizzate con i server di gioco ospitati su AmazonGameLift, inclusi i server in tempo reale. Puoi utilizzare la [GameLiftconsole Amazon](#) per la maggior parte delle attività di gestione delle risorse oppure puoi effettuare chiamate al servizio utilizzando lo strumento AWS Command Line Interface (AWS CLI) o l'AWSSDK.

Prepara i server di gioco per la distribuzione

Carica e configura il codice del server di gioco per prepararlo alla distribuzione e al 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 risorsa di build AmazonGameLift. Questa operazione, disponibile solo come AWS CLI comando, è il metodo più comune per caricare le build dei server di gioco.
- [CreateBuild](#)— Crea una nuova build utilizzando i file archiviati in un bucket Amazon S3.
- [ListBuilds](#)— Ottieni un elenco di tutte le build caricate in una GameLift regione Amazon.
- [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 da AmazonGameLift.

Gestione degli script di configurazione dei server in tempo reale

- [CreateScript](#)— Carica JavaScript file e crea una nuova risorsa di GameLift script Amazon.
- [ListScripts](#)— Ottieni un elenco di tutti gli script in tempo reale caricati in una regione AmazonGameLift.
- [DescribeScript](#)— Recupera le informazioni associate a uno script in tempo reale.
- [UpdateScript](#)— Modifica i metadati dello script e carica il contenuto dello script rivisto.
- [DeleteScript](#)— Rimuovi uno script in tempo reale da AmazonGameLift.

Configurazione delle risorse informatiche per l'hosting

Configura le risorse di hosting e distribuiscile con la build del tuo server di gioco o lo script di configurazione in tempo reale.

Crea e gestisci flotte

- [CreateFleet](#)— Configura e distribuisci una nuova GameLift flotta di risorse informatiche Amazon per gestire i tuoi server di gioco. Una volta distribuiti, i server di gioco vengono avviati automaticamente come configurati e pronti per ospitare sessioni di gioco.
- [ListFleets](#)— Ottieni un elenco di tutte le flotte in una GameLift regione amazzonica.

- [DeleteFleet](#)— Interrompi 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 sedi remote per una flotta e visualizza lo stato attuale di ciascuna sede.
 - [DeleteFleetLocations](#)— Rimuovi le sedi 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 una flotta.
 - [DescribeRuntimeConfiguration/UpdateRuntimeConfiguration](#)— Visualizza o modifica quali processi server (e quanti) eseguire su ciascuna istanza di un parco macchine.

Gestisci la capacità della flotta

- [DescribeEC2 InstanceLimits](#): recupera il numero massimo di istanze consentite per l'AWSaccount 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.
- Imposta il ridimensionamento automatico:
 - [PutScalingPolicy](#)— Attiva la scalabilità automatica basata sugli obiettivi o crea una politica di scalabilità automatica personalizzata o aggiorna una politica esistente.
 - [DescribeScalingPolicies](#)— Recupera una politica di scalabilità automatica esistente.
 - [DeleteScalingPolicy](#)— Eliminare una politica di scalabilità automatica e impedire che influisca sulla capacità di una flotta.
 - [StartFleetActions](#)— Riavvia le politiche di scalabilità automatica di una flotta.
 - [StopFleetActions](#)— Sospendere le politiche di ridimensionamento automatico di una flotta.

Monitorare l'attività del parco istanze.

- [DescribeFleetUtilization](#)— Recupera 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 sede 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, incluso il tempo di esecuzione di un gioco e il numero attuale di giocatori.

Impostare le code per un posizionamento ottimale 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 delle sessioni di gioco.
- [DescribeGameSessionQueues](#)— Recupera le code delle sessioni di gioco definite in una regione Amazon. GameLift
- [UpdateGameSessionQueue](#)— Cambia la configurazione della coda di una sessione di gioco.
- [DeleteGameSessionQueue](#)— Rimuovi una coda di sessioni 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](#)— Definire un nuovo alias e assegnarlo facoltativamente a una flotta.
- [ListAliases](#)— Ottieni tutti gli alias della flotta definiti in una GameLift regione Amazon.
- [DescribeAlias](#)— Recupera informazioni su un alias esistente.
- [UpdateAlias](#)— Modifica le impostazioni di un alias, ad esempio il reindirizzamento da una flotta all'altra.
- [DeleteAlias](#)— Rimuovere un alias dalla regione.
- [ResolveAlias](#)— Ottieni l'ID della flotta a cui fa riferimento un alias specificato.

Accesso alle istanze di hosting

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 di un parco istanze, inclusi ID istanza, indirizzo IP, posizione e stato.
- [GetInstanceAccess](#)— Richiedere le credenziali di accesso necessarie per connettersi in remoto a un'istanza specifica in un parco veicoli.

Configurazione del peering VPC

Crea e gestisci connessioni peering VPC tra le tue risorse di GameLift hosting Amazon e altre AWS risorse.

- [CreateVpcPeeringAuthorization](#)— Autorizza una connessione peering a uno dei tuoi VPC.
- [DescribeVpcPeeringAuthorizations](#)— Recupera autorizzazioni di connessione peering valide.
- [DeleteVpcPeeringAuthorization](#)— Eliminare un'autorizzazione alla connessione peering.
- [CreateVpcPeeringConnection](#)— Stabilisci una connessione peering tra il VPC di una GameLift flotta Amazon e uno dei tuoi VPC.
- [DescribeVpcPeeringConnections](#)— Recupera informazioni sulle connessioni peering VPC attive o in sospeso con una flotta Amazon. GameLift
- [DeleteVpcPeeringConnection](#)— Eliminare una connessione peering VPC con una flotta AmazonGameLift.

Inizia sessioni di gioco e unisciti ai giocatori

Richiama queste operazioni dal tuo servizio client di gioco per avviare nuove sessioni di gioco, ottenere informazioni sulle sessioni di gioco esistenti e unire i giocatori alle sessioni di gioco. Queste operazioni devono essere utilizzate con server di gioco personalizzati ospitati su AmazonGameLift. Se utilizzi server in tempo reale, gestisci le sessioni di gioco utilizzando il [Riferimento all'API client \(C#\) di Realtime Servers](#).

- Avviare nuove sessioni di gioco per uno o più giocatori.
 - [StartGameSessionPlacement](#)— Chiedi GameLift ad Amazon di trovare le migliori risorse di hosting disponibili e 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 monitorare la disponibilità dell'hosting in più regioni e utilizza gli algoritmi FleetIQ per dare priorità ai posizionamenti in base alla latenza dei giocatori, al costo di hosting, alla posizione, ecc.

- [DescribeGameSessionPlacement](#)— Ottieni dettagli e stato su una richiesta di collocamento.
- [StopGameSessionPlacement](#)— Annulla una richiesta di collocamento.
- [CreateGameSession](#)— Inizia una nuova sessione di gioco vuota in una posizione specifica della flotta. Questa operazione consente un maggiore controllo su dove iniziare la sessione di gioco, anziché utilizzare FleetIQ per valutare le opzioni di posizionamento. Devi aggiungere giocatori alla nuova sessione di gioco in un passaggio separato.
- Fai partecipare i giocatori alle sessioni di gioco esistenti. Trova sessioni di gioco in corso con slot disponibili e riservale per i nuovi giocatori.
 - [CreatePlayerSession](#)— Prenota uno slot aperto per consentire a un giocatore di partecipare a una sessione di gioco.
 - [CreatePlayerSessions](#)— Prenota 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, incluso il periodo di tempo attivo e il numero attuale dei 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, inclusi lo stato, il tempo di gioco e i dati del giocatore.
 - [UpdateGameSession](#)— Modifica le impostazioni della sessione di gioco, come il numero massimo di giocatori e la politica di partecipazione.
 - [GetGameSessionLogUrl](#)— Ottieni la posizione dei log salvati per una sessione di gioco.

Riferimento Amazon GameLift Realtime Servers

Questa sezione contiene la documentazione di riferimento per l'SDK Amazon GameLift Realtime Servers. Include l'API Realtime Client e una guida per la configurazione dello script Realtime Servers.

Argomenti

- [Riferimento all'API client \(C#\) di Realtime Servers](#)
- [Riferimento allo script Amazon GameLift Realtime Servers](#)

Riferimento all'API client (C#) di Realtime Servers

Usa l'API Realtime Client per preparare i tuoi client di gioco multiplayer da utilizzare con Amazon GameLift Realtime Servers. Per i dettagli sul processo di integrazione, consulta [Prepara il tuo server in tempo reale](#). L'API client contiene una serie di chiamate API sincrone e callback asincroni che consentono a un client di gioco di connettersi a un server in tempo reale e scambiare messaggi e dati con altri client di gioco tramite il server.

Questa API è definita nelle seguenti librerie:

Client.cs

- [Operazioni sincrone](#)
- [Chiamate asincrone](#)
- [Tipi di dati](#)

Per configurare l'API del client Realtime

1. Scarica l'[SDK del client Amazon GameLift Realtime](#).
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 Realtime Client al tuo progetto client di gioco.

Riferimento all'API client (C#) di Realtime Servers: Actions

Questo riferimento all'API C# Realtime Client può aiutarti a preparare il tuo gioco multiplayer per l'uso con i server Realtime distribuiti sulle flotte Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Prepara il tuo server in tempo reale](#).

- 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 `AWSSDK` dell'GameLiftAPI Amazon [StartGameSessionPlacementCreateGameSession](#), oppure. [DescribeGameSessions](#)

Note

Se il server Realtime è in esecuzione su un parco istanze con un certificato TLS, è necessario utilizzare il nome DNS.

Tipo: String

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 [StartGameSessionPlacementCreateGameSessionDescribeGameSession](#) richiesta.

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 [ConnectionStatus](#)enum 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. Gli ID di gruppo sono definiti dagli sviluppatori.

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. Gli ID di gruppo sono definiti dagli sviluppatori.

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. Gli ID di gruppo sono definiti dagli sviluppatori.

Tipo: integer

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

Riferimento API client (C#) di Realtime Servers: callback asincroni

Usa questo riferimento all'API C# Realtime Client per aiutarti a preparare il tuo gioco multiplayer da utilizzare con i server Realtime distribuiti sulle flotte Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Prepara il tuo server in tempo reale](#).

- [Operazioni sincrone](#)
- Chiamate asincrone
- [Tipi di dati](#)

Un client di gioco deve implementare questi metodi di chiamata per rispondere a eventi. Il server Realtime richiama questi callback per inviare informazioni relative al gioco al client di gioco. I callback

per gli stessi eventi possono anche essere implementati con una logica di gioco personalizzata nello script del server Realtime. Consultare [Richiamate di script per server in tempo reale](#).

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()

Richiamato 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

dataReceivedEventArgs

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

groupMembershipEventArti

Le informazioni relative all'attività di appartenenza al gruppo.

Tipo: [GroupMembershipEventArgs](#)

Campo obbligatorio: sì

Valore restituito

Questo metodo non restituisce nulla.

Riferimento API client (C#) di Realtime Servers: tipi di dati

Questo riferimento all'API C# Realtime Client può aiutarti a preparare il tuo gioco multiplayer per l'uso con i server Realtime distribuiti sulle flotte Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Prepara il tuo server in tempo reale](#).

- [Operazioni sincrone](#)
- [Chiamate asincrone](#)
- Tipi di dati

ClientConfiguration

Informazioni su come il client di gioco si connette a un server in tempo reale.

Indice

ConnectionType

Tipo di connessione client/server da utilizzare, protetta o non protetta. Se non specifichi un tipo di connessione, l'impostazione predefinita corrisponde a una connessione non protetta.

Note

Per la connessione a un server Realtime su un parco istanze protetto 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 client di gioco e/o sul giocatore che richiede una connessione con un server in tempo reale.

Indice

playerSessionId

ID univoco rilasciato da Amazon GameLift quando viene creata una nuova sessione giocatore. Un ID di sessione del giocatore è specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLiftAPI [StartGameSessionPlacementCreateGameSession](#), [DescribeGameSessionPlacement](#), o [DescribePlayerSessions](#).

Tipo: String

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 devono essere elaborate dallo script del server Realtime 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. I messaggi creati utilizzando `NewMessage()` hanno già il codice operativo impostato, ma è possibile modificarlo 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. Gli ID di gruppo sono definiti dagli sviluppatori.

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. Gli ID di gruppo sono definiti dagli sviluppatori.

Tipo: integer

Campo obbligatorio: sì

playerId

Elenco di ID dei giocatori che sono attualmente membri del gruppo specificato.

Tipo: array di numeri interi

Campo obbligatorio: sì

enumerazioni;

Le enumerazioni definite per Realtime Client SDK sono definite come segue:

ConnectionStatus

- **CONNESSO**: il client di gioco è connesso al server Realtime solo con 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 server Realtime con 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 Realtime con una connessione TCP e UDP. 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 Realtime in risposta a una richiesta del client di gioco. [Disconnect\(\)](#)
- **DISCONNESSO**: il client di gioco è stato disconnesso dal server Realtime per un motivo diverso da una chiamata di disconnessione del client.

ConnectionType

- **RT_OVER_WSS_DTLS_TLS12** — Tipo di connessione sicura.

Da utilizzare con server in tempo reale in esecuzione su una GameLift flotta con un certificato TLS generato. 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 sicura.
- **RT_OVER_WEBSOCKET** — Tipo di connessione non sicuro. Questo valore non è più preferito.

DeliveryIntent

- **VELOCE**: fornito utilizzando un canale UDP.
- **AFFIDABILE**: fornito tramite una connessione TCP.

Riferimento allo script Amazon GameLift Realtime Servers

Usa queste risorse per creare una logica personalizzata nei tuoi script in tempo reale.

Argomenti

- [Richiamate di script per server in tempo reale](#)
- [Interfaccia Realtime Servers](#)

Richiamate di script per server in tempo reale

Puoi fornire una logica personalizzata per rispondere agli eventi implementando questi callback nel tuo script Realtime.

Init

Inizializza il server Realtime e riceve un'interfaccia server in tempo reale.

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 interrotto dal servizio AmazonGameLift. 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)
```

Interfaccia Realtime Servers

Quando uno script in tempo reale viene inizializzato, viene restituita un'interfaccia per il server Realtime. In questo argomento vengono illustrati i metodi e le proprietà disponibili tramite l'interfaccia. Scopri di più sulla scrittura di script in tempo reale e visualizza un esempio di script dettagliato in.

[Creazione di uno script in tempo reale](#)

L'interfaccia Realtime fornisce l'accesso ai seguenti oggetti:

- session
- player (giocatore)
- gameMessage
- configurazione

Oggetto Realtime Session

Utilizza questi metodi per accedere alle informazioni relative al server ed eseguire operazioni correlate al server.

`getPlayers()`

Recupera un elenco di ID peer per i giocatori attualmente connessi a sessioni di gioco. Restituisce un array di oggetti player.

Sintassi

```
rtSession.getPlayers()
```

`broadcastGroupMembershipAggiornamento ()`

Attiva la distribuzione di un elenco aggiornato delle appartenenze ai gruppi di giocatori. Specifica quale appartenenza trasmettere (`groupIdToBroadcast`) e il gruppo a cui inviare l'aggiornamento (`targetGroupId`). Gli ID dei gruppi devono essere un numero intero positivo o «-1» per indicare tutti i gruppi. Vedi un [Esempio di script Realtime Servers](#) esempio di ID di gruppo definiti dall'utente.

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 richiamata dallo script Realtime per uscire senza problemi da una 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 utilizzando `newTextGameMessage` o `newBinaryGameMessage`, dal server Realtime al destinatario di un giocatore utilizzando il canale UDP. Identifica il destinatario utilizzando l'ID peer del giocatore.

Sintassi

```
rtSession.sendMessage(gameMessage, targetPlayer)
```

sendGroupMessage()

Invia un messaggio, creato utilizzando `newTextGameMessage` o `newBinaryGameMessage`, dal server Realtime a tutti i giocatori di un gruppo di giocatori utilizzando il canale UDP. Gli ID dei gruppi devono essere un numero intero positivo o «-1» per indicare tutti i gruppi. Vedi un [Esempio di script Realtime Servers](#) esempio di ID di gruppo definiti dall'utente.

Sintassi

```
rtSession.sendGroupMessage(gameMessage, targetGroup)
```

sendReliableMessage()

Invia un messaggio, creato utilizzando `newTextGameMessage` o `newBinaryGameMessage`, dal server Realtime al destinatario di un giocatore utilizzando il canale TCP. Identifica il destinatario utilizzando l'ID peer del giocatore.

Sintassi

```
rtSession.sendReliableMessage(gameMessage, targetPlayer)
```

sendReliableGroupMessaggio ()

Invia un messaggio, creato utilizzando `newTextGameMessage` o `newBinaryGameMessage`, dal server Realtime a tutti i giocatori di un gruppo di giocatori utilizzando il canale TCP. ID di gruppo che

devono essere un numero intero positivo o «-1" per indicare tutti i gruppi. Vedi un [Esempio di script Realtime Servers](#) esempio di ID di gruppo definiti dall'utente.

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 Realtime (vedi [RTMessage](#)). 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 Realtime (vedi [RTMessage](#)). Restituisce un oggetto gameMessage.

Sintassi

```
rtSession.newBinaryGameMessage(opcode, sender, binaryPayload)
```

Oggetto player

Accedi alle informazioni sul giocatore.

player.peerId

ID univoco assegnato a un client di gioco quando si connette al server Realtime e partecipa alla sessione di gioco.

giocatore. playerSessionId

ID della sessione del giocatore a cui il client di gioco ha fatto riferimento quando si è connesso al server Realtime e si è unito alla sessione di gioco.

Oggetto game message

Utilizza questi metodi per accedere ai messaggi ricevuti dal server Realtime. I messaggi ricevuti dai client di gioco hanno la struttura [RTMessage](#).

getPayloadAsText ()

Ottiene il payload del messaggio 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 utilizzato per sovrascrivere le configurazioni predefinite.

Configurazione. Max Players

Il numero massimo di connessioni client/server che possono essere accettate daRealTimeServers.

L'impostazione predefinita 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.

Riferimento SDK per GameLift server Amazon

Questa sezione contiene la documentazione di riferimento per l'Amazon GameLift Server SDK. Usa il server SDK per integrare i tuoi server di gioco personalizzati per comunicare con il GameLift servizio Amazon.

Argomenti

- [Riferimento SDK per GameLift server Amazon per C++](#)
- [Riferimento SDK per GameLift server Amazon per C#](#)
- [Riferimento SDK GameLift del server Amazon per Go](#)
- [Riferimento SDK GameLift del server Amazon per Unreal Engine](#)

Riferimento SDK per GameLift server Amazon per C++

Puoi usare questo riferimento all'SDK per server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento Amazon GameLift Server SDK 5.x per C++](#)
- [Riferimento all'SDK 3.x per server Amazon GameLift C++](#)

Riferimento Amazon GameLift Server SDK 5.x per C++

Questa guida di riferimento per Amazon GameLift C++ Server SDK 5.x può aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Note

Questo argomento descrive l'API Amazon GameLift C++ che puoi usare quando crei con la libreria standard C++ (`std`). In particolare, questa documentazione si applica al codice compilato con l'opzione `-DDGAMELIFT_USE_STD=1`.

Argomenti

- [Riferimento Amazon GameLift Server SDK \(C++\) 5.x: azioni](#)
- [Riferimento all'SDK per GameLift server Amazon \(C++\): tipi di dati](#)

Riferimento Amazon GameLift Server SDK (C++) 5.x: azioni

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per dettagli sul processo di integrazione, consulta. [Aggiungi Amazon GameLift al tuo server di gioco](#)

Note

Questo argomento descrive l'API Amazon GameLift C++ che puoi usare quando crei con la C++ Standard Library (`std`). In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DDGAMELIFT_USE_STD=1`

Azioni

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessReadyAsync\(\)](#)
- [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

```
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'Amazon GameLift SDK per una flotta EC2 gestita. Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo legge i parametri del server dall'ambiente host per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK();
```

Valore restituito

Restituisce un [the section called "InitSDK Output"](#) oggetto che indica se il processo del server è pronto per la chiamata. [ProcessReady\(\)](#)

Esempio

```
//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
    Aws::GameLift::Server::InitSDK();
```

InitSDK()

Inizializza l'Amazon GameLift SDK per una flotta. Anywhere Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo richiede parametri espliciti del server per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK(serverParameters);
```

Parametri

[ServerParameters](#)

Per inizializzare un server di gioco su una GameLift Anywhere flotta Amazon, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket usato per connetterti al tuo 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 GameLift flotta Amazon contenente i tuoi dati di GameLift Anywhere calcolo Amazon.
- Il token di autorizzazione generato dall' GameLift operazione Amazon.

Valore restituito

Restituisce un [the section called “InitSDK Output”](#) 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

GameLift Anywhere Esempio di Amazon

```
//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 GameLift agent to enable
    further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
    Aws::GameLift::Server::InitSDK(serverParameters);
```

ProcessReady()

Notifica ad Amazon GameLift 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 GameLift servizio Amazon 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 che Amazon GameLift acquisisca e archivi.

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 GameLift servizio Amazon che il processo del server è pronto per ospitare sessioni di gioco. Questo metodo deve 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 che Amazon GameLift può 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 GameLift servizio Amazon 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 che Amazon GameLift acquisisca e archivi.

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 ad Amazon GameLift che il processo del server sta terminando. Richiama 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 ad Amazon GameLift che il processo del server ha attivato una sessione di gioco ed è ora pronto per ricevere le connessioni dei giocatori. Questa azione deve essere richiamata come parte della funzione di `onStartGameSession()` 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. Amazon GameLift chiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto ad Amazon GameLift.
- 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 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.

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. [Rispondere a una notifica di arresto del processo del server](#)

Esempio

```
Aws::GameLift::AwsLongOutcome TermTimeOutcome =  
    Aws::GameLift::Server::GetTerminationTime();
```


AcceptPlayerSession()

Notifica ad Amazon GameLift che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID di sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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 gestisce una richiesta di connessione che include la convalida e il rifiuto di ID di sessione giocatore non validi.

```
void ReceiveConnectingPlayerSessionID (Connection& connection, const std::string&
playerSessionId)
{
    Aws::GameLift::GenericOutcome connectOutcome =
    Aws::GameLift::Server::AcceptPlayerSession(playerSessionId);
    if(connectOutcome.IsSuccess())
    {
        connectionToSessionMap.emplace(connection, playerId);
        connection.Accept();
    }
    else
    {
        connection.Reject(connectOutcome.GetError().GetMessage());
    }
}
```

RemovePlayerSession()

Notifica ad Amazon GameLift che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift cambia lo slot del giocatore rendendolo disponibile.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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

```
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 NextToken e impostando il valore Limite su 10, Amazon GameLift restituisce i primi 10 record delle sessioni di gioco 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 [FlexMatch riempimento](#).

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift fornisce dati aggiornati sui 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, Amazon ne GameLift 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 tua GameLift Anywhere risorsa di elaborazione Amazon e Amazon. GameLift Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una GameLift Anywhere flotta Amazon. Per ulteriori informazioni, consulta, [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 GameLift Amazon 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 fornite un nome per la sessione di ruolo, viene utilizzato il valore predefinito "*[fleet-id] - [host-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 GameLift gioco Amazon dalla memoria. Come best practice, richiama 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 reinizializza prima di notificare `InitSDK()` ad Amazon GameLift che il processo è pronto per ospitare una sessione di gioco con. `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);
}
```

Riferimento all'SDK per GameLift server Amazon (C++): tipi di dati

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per dettagli sul processo di integrazione, consulta. [Aggiungi Amazon GameLift al tuo server di gioco](#)

Note

Questo argomento descrive l'API Amazon GameLift C++ che puoi usare quando crei con la C++ Standard Library (`std`). In particolare, questa documentazione si applica al codice compilato con l'opzione. `-DDGAMELIFT_USE_STD=1`

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](#)
- [InitSDK Output](#)
- [GameLiftError](#)
- [enumerazioni;](#)

LogParameters

Un oggetto che identifica i file generati durante una sessione di gioco che desideri che Amazon GameLift carichi e archivi al termine della sessione di gioco. Il server di gioco fornisce `LogParameters` ad Amazon GameLift come parte di un `ProcessParameters` oggetto in una [ProcessReady\(\)](#) chiamata.

Proprietà	Descrizione
-----------	-------------

LogPaths	<p>L'elenco dei percorsi di directory dei file di registro del server di gioco che desideri che Amazon GameLift memorizzi 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.</p> <p>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 <code>MyGame\sessionLogs\</code>, il percorso si troverebbe <code>c:\game\MyGame\sessionLogs</code> su un'istanza di Windows.</p> <p>Tipo: <code>std::vector<std::string></code></p> <p>Required: No</p>
----------	---

ProcessParameters

Questo tipo di dati contiene il set di parametri inviati ad Amazon GameLift in un [ProcessReady\(\)](#).

Proprietà	Descrizione
LogParameters	<p>Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift copia e archivia i file per accessi futuri.</p> <p>Tipo: <code>Aws::GameLift::Server:: LogParameters</code></p> <p>Required: No</p>
OnHealthCheck	<p>La funzione di callback GameLift richiamata da Amazon per richiedere un rapporto sullo</p>

stato di salute del processo del server. Amazon GameLift chiama 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 registra il processo del server come non integro.

Tipo: `std::function<bool()>`
`onHealthCheck`

Required: No

OnProcessTerminate

La funzione di callback GameLift richiamata da Amazon per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift 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

OnRefreshConnection

Il nome della funzione di callback richiamata da Amazon GameLift per aggiornare la connessione con il server di gioco.

Tipo: `void OnRefreshConnectionDelegate()`

Campo obbligatorio: sì

OnStartGameSession

La funzione di callback che Amazon GameLift richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta a una richiesta del cliente [CreateGameSession](#). La funzione di callback passa un [GameSession](#) oggetto, come definito nell'Amazon GameLift API Reference.

```
Tipo: const std::function<void  
(Aws::GameLift::Model::Game  
Session)> onStartGameSession
```

Campo obbligatorio: sì

OnUpdateGameSession

La funzione di callback GameLift richiamata da Amazon per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift chiama questa funzione quando viene 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::Gam  
eLift::Server::Model::Updat  
eGameSession)> onUpdateG  
ameSession
```

Required: No

Porta	<p>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.</p> <p>Tipo: Integer</p> <p>Campo obbligatorio: sì</p>
--------------	---

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	<p>Un GameSession oggetto definito dall' GameLift API Amazon. L'GameSession oggetto contiene proprietà che descrivono una sessione di gioco.</p> <p>Tipo: <code>Aws::GameLift::Server::GameSession</code></p> <p>Campo obbligatorio: sì</p>
UpdateReason	<p>Il motivo per cui la sessione di gioco viene aggiornata.</p> <p>Tipo: <code>Aws::GameLift::Server::UpdateReason</code></p>

Proprietà	Descrizione
	Campo obbligatorio: sì
BackfillTicketId	L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco. Tipo: <code>std::string</code> 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: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> Tipo: <code>std::string</code> Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: <code>std::string</code> Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco. Tipo: <code>std::string</code> Required: No

Proprietà	Descrizione
MaximumPlayerSessionCount	<p>Il numero massimo di connessioni dei giocatori alla sessione di gioco.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>
Porta	<p>Il numero di porta per la sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta.</p> <p>Tipo: <code>in</code></p> <p>Required: No</p>
IpAddress	<p>L'indirizzo IP della sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
GameSessionData	<p>Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

Proprietà	Descrizione
MatchmakerData	<p>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.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
GameProperties	<p>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.</p> <p>Tipo: <code>std::vector < GameProperty ></code></p> <p>Required: No</p>

Proprietà	Descrizione
DnsName	<p>L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato:</p> <ul style="list-style-type: none"> • Flotte abilitate per TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> • Flotte non compatibili con TLS: <code>ec2-unique identifier>.compute.amazonaws.com</code> <p>Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi utilizzare il nome DNS, non l'indirizzo IP.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

ServerParameters

Informazioni utilizzate per mantenere la connessione tra il server di gioco su una GameLift Anywhere flotta Amazon e il GameLift servizio Amazon. Queste informazioni vengono utilizzate quando si avviano nuovi processi server con [InitSDK\(\)](#). Per i server ospitati su istanze EC2 GameLift gestite da Amazon, usa un oggetto vuoto.

Proprietà	Descrizione
webSocketUrl	<p>GameLiftServerSdkEndpoint Amazon GameLift restituisce quando RegisterCompute richiedi una risorsa di GameLift Anywhere elaborazione Amazon.</p> <p>Tipo: <code>std::string</code></p>

Proprietà	Descrizione
	Campo obbligatorio: sì
ID del processo	<p>Un identificatore univoco registrato nel processo del server che ospita il gioco.</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>
hostId	<p>HostIDViene ComputeName utilizzato al momento della registrazione del computer. Per ulteriori informazioni, vedere, RegisterCompute.</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>
FleetID	<p>L'identificatore univoco del parco macchine su cui è registrato il computer. Per ulteriori informazioni, vedere, RegisterCompute</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>
AuthToken	<p>Il token di autenticazione generato da Amazon GameLift che autentica il tuo server su Amazon. GameLift Per ulteriori informazioni, consulta, GetComputeAuthToken</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di riempimento del matchmaking. Il server di gioco comunica queste informazioni ad Amazon GameLift tramite una [StartMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>Un identificatore di sessione di gioco univoco. L'operazione API GetGameSessionId restituisce l'identificatore in formato ARN.</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>
MatchmakingConfigurationArn	<p>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 <code>matchmaker data</code>. Scopri di più sui dati dei matchmaker in Work with matchmaker data.</p> <p>Tipo: <code>std::string</code></p> <p>Campo obbligatorio: sì</p>
Players	<p>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 rappresentino un buon abbinamento per i giocatori attuali.</p> <p>Tipo: <code>std::vector<Player></code></p> <p>Campo obbligatorio: sì</p>
TicketId	<p>Un identificatore univoco per un ticket di richiesta di matchmaking o match backfill. Se non fornisci un valore, Amazon ne GameLift genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill</p>

Proprietà	Descrizione
	<p>degli abbinamenti o annullare la richiesta, se necessario.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

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 aggiunge le informazioni sulla squadra dopo la fine di una partita.

Proprietà	Descrizione
LatencyInSM	<p>Un insieme di valori espressi in millisecondi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione.</p> <p>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.</p> <p>Tipo: <code>Dictionary<string,int></code></p> <p>Required: No</p>
PlayerAttributes	<p>Una raccolta di coppie chiave:valore contenenti informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a quelle <code>PlayerAttributes</code> utilizzate in un set di regole di matchmaking.</p>

Proprietà	Descrizione
	<p>Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue.</p> <p>Tipo: <code>std::map<std::string, AttributeValue></code></p> <p>Required: No</p>
PlayerId	<p>Un identificatore univoco per un giocatore.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
Team	<p>Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

DescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo del server fornisce queste informazioni con una [DescribePlayerSessions\(\)](#) chiamata ad Amazon GameLift.

Proprietà	Descrizione
GameSessionId	<p>Un identificatore di sessione di gioco univoco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata.</p> <p>Il formato dell'ID della sessione di gioco è <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID</code></p>

Proprietà	Descrizione
	<p><code>string>. GameSessionID</code> È una stringa ID personalizzata o un</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
PlayerSessionId	<p>L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
PlayerId	<p>L'identificatore univoco di un giocatore. Usa questo parametro per richiedere tutte le sessioni di gioco per un giocatore specifico. Per informazioni, consulta Genera gli ID dei giocatori.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

Proprietà	Descrizione
PlayerSessionStatusFilter	<p>Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono:</p> <ul style="list-style-type: none">• 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). <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
NextToken	<p>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.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

Proprietà	Descrizione
Limite	<p>Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al GameLift servizio Amazon tramite una [StopMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>Un identificatore univoco della sessione di gioco della richiesta annullata.</p> <p>Tipo: <code>char []</code></p> <p>Required: No</p>
MatchmakingConfigurationArn	<p>Un identificatore univoco del matchmaker a cui è stata inviata questa richiesta.</p> <p>Tipo: <code>char []</code></p> <p>Required: No</p>
TicketId	<p>Un identificatore univoco del ticket di richiesta di riempimento da annullare.</p> <p>Tipo: <code>char []</code></p> <p>Required: No</p>

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 deve utilizzare esattamente una delle proprietà disponibili: SN,SL, oSDM.

Proprietà	Descrizione
AttrType	<p>Specifica il tipo di valore dell'attributo. I possibili tipi di valori degli attributi includono:</p> <ul style="list-style-type: none"> • NONE • STRINGA • DOPPIA • ELENCO_STRINGHE • STRING_DOUBLE_MAP <p>Required: No</p>
S	<p>Rappresenta un valore di attributo di tipo stringa.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
N	<p>Rappresenta un valore di attributo numerico.</p> <p>Tipo: <code>double</code></p> <p>Required: No</p>
SL	<p>Rappresenta una matrice di valori di attributi di tipo stringa.</p> <p>Tipo: <code>std::vector<std::string></code></p> <p>Required: No</p>

Proprietà	Descrizione
SDM	<p>Rappresenta un dizionario di chiavi di stringa e valori doppi.</p> <p>Tipo: <code>std::map<std::string, double></code></p> <p>Required: No</p>

GetFleetRoleCredentialsRequest

Questo tipo di dati offre al server di gioco un accesso limitato alle altre risorse. AWS Per ulteriori informazioni, consulta [Configura un ruolo di servizio IAM per Amazon GameLift](#).

Proprietà	Descrizione
RoleArn	<p>L'Amazon Resource Name (ARN) del ruolo di servizio che estende l'accesso limitato alle tue AWS risorse.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
RoleSessionName	<p>Il nome della sessione di ruolo che puoi utilizzare e per identificare in modo univoco una AWS Security Token Service AssumeRole sessione. Questo nome è esposto nei log di controllo, come quelli di CloudTrail</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

AwsLongOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: long</p> <p>Required: No</p>
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: long&&</p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

AwsStringOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: std::string</p> <p>Required: No</p>

Proprietà	Descrizione
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: long&&</p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

DescribePlayerSessionsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “DescribePlayerSessionsResult”</p> <p>Required: No</p>
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p>

Proprietà	Descrizione
	Tipo: <code>Aws::GameLift::Server::Model::DescribePlayerSessionsResult</code> Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> 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

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 a pagina sequenziale 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: <code>std::string</code> Campo obbligatorio: sì

Proprietà	Descrizione
PlayerSessions	Tipo: <code>IList<the section called "PlayerSession"></code> Campo obbligatorio:
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: <code>std::string&&</code> Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> 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: <code>bool</code> Campo obbligatorio: sì

Proprietà	Descrizione
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

GenericOutcomeCallable

Questo tipo di dati è un risultato generico asincrono. Possiede le seguenti proprietà:

Proprietà	Descrizione
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

PlayerSession

Questo tipo di dati rappresenta una sessione di gioco che Amazon GameLift trasmette al server di gioco. Per ulteriori informazioni, consulta [PlayerSession](#).

Proprietà	Descrizione
CreationTime	<p>Tipo: long</p> <p>Required: No</p>

Proprietà	Descrizione
FleetId	Tipo: <code>std::string</code> Required: No
GameSessionId	Tipo: <code>std::string</code> Required: No
IpAddress	Tipo: <code>std::string</code> Required: No
PlayerData	Tipo: <code>std::string</code> Required: No
PlayerId	Tipo: <code>std::string</code> Required: No
PlayerSessionId	Tipo: <code>std::string</code> Required: No
Porta	Tipo: <code>int</code> Required: No

Proprietà	Descrizione
Stato	<p>Stato sessione giocatore su cui filtrare i risultati . Quando PlayerId viene fornito un PlayerSessionId or, non PlayerSessionStatusFilter ha alcun effetto sulla risposta.</p> <p>Tipo: Un PlayerSessionStatus enum. I valori possibili sono:</p> <ul style="list-style-type: none"> • ACTIVE • COMPLETED • NOT_SET • RISERVATO • TIMEOUT <p>Required: No</p>
TerminationTime	<p>Tipo: long</p> <p>Required: No</p>
DnsName	<p>Tipo: std::string</p> <p>Required: No</p>

StartMatchBackfillOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “StartMatchBackfillIResult”</p>

Proprietà	Descrizione
	Required: No
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: <code>StartMatchBackfillResult&&</code></p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

StartMatchBackfillResult

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
TicketId	<p>Un identificatore univoco per un ticket di matchmaking. Se qui non viene specificato alcun ID del ticket, Amazon ne GameLift genererà uno sotto forma di UUID. Usa questo identificatore per tenere traccia dello stato del ticket di backfill della partita e recuperare i risultati delle partite.</p> <p>Tipo: <code>std::string</code></p>

Proprietà	Descrizione
	Required: No

GetComputeCertificateOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “GetComputeCertificateResult”</p> <p>Required: No</p>
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: <code>Aws::GameLift::Server::Model::GetComputeCertificateResult&&</code></p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

GetComputeCertificateResult

Il percorso del certificato TLS sul computer e il nome host del computer.

Proprietà	Descrizione
CertificatePath	<p>Il percorso del certificato TLS sulla tua risorsa di elaborazione. Quando si utilizza una flotta GameLift gestita da Amazon, questo percorso contiene:</p> <ul style="list-style-type: none"> • <code>certificate.pem</code> : Il certificato dell'utente finale. La catena completa di certificati è la combinazione dei certificati <code>certificateChain.pem</code> aggiunti a questo certificato. • <code>certificateChain.pem</code> : La catena di certificati che contiene il certificato principale e i certificati intermedi. • <code>rootCertificate.pem</code> : Il certificato principale. • <code>privateKey.pem</code> : la chiave privata per il certificato dell'utente finale. <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
ComputeName	<p>Il nome della risorsa di calcolo.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

GetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “GetFleetRoleCredentialsResult”</p> <p>Required: No</p>
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: <code>Aws::GameLift::Server::Model::GetFleetRoleCredentialsResult</code></p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

GetFleetRoleCredentialsResult

Proprietà	Descrizione
AccessKeyId	<p>L'ID della chiave di accesso per autenticare e fornire l'accesso alle tue risorse. AWS</p> <p>Tipo: <code>string</code></p>

Proprietà	Descrizione
	Required: No
AssumedRoleId	L'ID dell'utente a cui appartiene il ruolo di servizio. Tipo: <code>string</code> Required: No
AssumedRoleUserArn	L'Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio. Tipo: <code>string</code> Required: No
Expiration	Il periodo di tempo che manca alla scadenza delle credenziali di sessione. Tipo: <code>DateTime</code> Required: No
SecretAccessKey	L'ID della chiave di accesso segreta per l'autenticazione. Tipo: <code>string</code> Required: No
SessionToken	Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse. Tipo: <code>string</code> Required: No

Proprietà	Descrizione
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

InitSDK Output

Note

InitSDKOut come 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	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

GameLiftError

Proprietà	Descrizione
ErrorType	<p>Il tipo di errore.</p> <p>Tipo: Un enum. GameLiftErrorType</p> <p>Required: No</p>
ErrorMessage	<p>Messaggio di errore.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>
ErrorName	<p>Il nome dell'errore.</p> <p>Tipo: <code>std::string</code></p> <p>Required: No</p>

enumerazioni;

Le enumerazioni definite per l'Amazon GameLift server SDK (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 ad Amazon non è riuscita. GameLift
- `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_INVALID_`

- `WEBSOCKET_CONNECT_FAILURE_TIMEOUT`
- `ALREADY_INITIALIZED` — Il server o il client GameLift Amazon è già stato inizializzato con `Initialize ()`.
- `FLEET_MISMATCH` — La flotta di destinazione non corrisponde alla flotta di una `GameSession` o `PlayerSession`.
- `GAMELIFT_CLIENT_NOT_INITIALIZED` — Il client Amazon non è stato inizializzato. GameLift
- `GAMELIFT_SERVER_NOT_INITIALIZED` — Il server Amazon non è stato inizializzato. GameLift
- `GAME_SESSION_ENDED_FAILED` — L'SDK di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco.
- `GAME_SESSION_NOT_READY` — La sessione di gioco di GameLift Amazon Server non è stata attivata.
- `GAME_SESSION_READY_FAILED` — L'SDK di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare che la sessione di gioco è pronta.
- `INITIALIZATION_MISMATCH` — È stato chiamato un metodo client dopo Server: `:Initialize ()` o viceversa.
- `NOT_INITIALIZED` — Il GameLift server o il client Amazon non è stato inizializzato 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 di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare la fine del processo.
- `PROCESS_NOT_ACTIVE` — Il processo del server non è ancora attivo, non è legato 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 di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare che il processo è pronto.
- `SDK_VERSION_DETECTION_FAILED` — Rilevamento della versione SDK non riuscito.
- `STX_CALL_FAILED` — Una chiamata al componente backend del server `xSTx` non è riuscita.
- `STX_INITIALIZATION_FAILED` — L'inizializzazione del componente di backend del server `XStx` non è riuscita.
- `UNEXPECTED_PLAYER_SESSION` — Il server ha rilevato una sessione 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.

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.

Riferimento all'SDK 3.x per server Amazon GameLift C++

Puoi usare questo riferimento all'SDK 3.x per server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento Amazon GameLift Server SDK \(C++\): azioni](#)
- [Riferimento Amazon GameLift Server SDK \(C++\): tipi di dati](#)

Riferimento Amazon GameLift Server SDK (C++): azioni

Puoi usare questo riferimento all'SDK per server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Operazioni

- [AcceptPlayerSession\(\)](#)
- [ActivateGameSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetInstanceCertificate\(\)](#)
- [GetSdkVersion\(\)](#)
- [GetTerminationTime\(\)](#)
- [InitSDK\(\)](#)
- [ProcessEnding\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessReadyAsync\(\)](#)
- [RemovePlayerSession\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [TerminateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [Distruggi \(\)](#)

AcceptPlayerSession()

Notifica al GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID della sessione del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot per il giocatore nella sessione di gioco. Una volta convalidato, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO a ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(const std::string& playerSessionId);
```

Parametri

playerSessionId

ID univoco emesso dal GameLift servizio Amazon in risposta a una chiamata all'azione [CreatePlayerSession](#) dell'GameLiftAPI Amazon 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 composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio illustra una funzione per gestire una richiesta di connessione, inclusi la convalida e il rifiuto di ID di sessioni giocatore non validi.

```
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 GameLift servizio Amazon che il processo del server ha avviato una sessione di gioco ed è ora 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 `onStartGameSession()`.

```
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 [PlayerSession](#) dati dell'GameLiftAPI Amazon AWS SDK.

Esempio

Questo esempio illustra una richiesta per tutte le sessioni giocatore attivamente connesse a una sessione di gioco specificata. Omettendo `NextToken` e impostando il `Limit` valore su 10, Amazon GameLift restituisce i record delle sessioni dei primi 10 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);
```

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 al parco veicoli e alle relative istanze. AWS Certificate Manager genera questo certificato quando si crea una nuova flotta 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 certificati TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato principale, 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 [GetCertificateResponse Elements](#) in the 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 del server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata dal GameLift servizio Amazon. [Amazon GameLift può chiamare `onProcessTerminate\(\)` per i seguenti motivi: \(1\) quando il processo del server ha segnalato problemi di salute o non ha risposto ad AmazonGameLift, \(2\) quando si interrompe l'istanza durante un evento di ridimensionamento o \(3\) quando un'istanza viene interrotta a causa di un'interruzione 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

```
AwsLongOutcome GetTerminationTime();
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

In caso di successo, restituisce l'ora di terminazione come `AwsLongOutcome` come oggetto. Il valore è il tempo di terminazione, espresso in zecche trascorse da 0001 00:00:00. Ad esempio, il valore data/ora 2020-09-13 12:26:40 -000Z è uguale a 637355968000000000 zecche. 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'Amazon GameLift SDK. Questo metodo deve essere richiamato all'avvio, prima che si verifichi qualsiasi altra inizializzazione GameLift relativa ad Amazon.

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 chiamata [ProcessReady\(\)](#).

Esempio

```
Aws::GameLift::Server::InitSDKOutcome initOutcome =  
    Aws::GameLift::Server::InitSDK();
```

ProcessEnding()

Notifica al GameLift servizio Amazon che il processo del server si sta spegnendo. 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 che il metodo esce con un codice 0, puoi terminare il processo con un codice di uscita riuscito. Puoi anche uscire dal processo con un codice di errore. Se esci con un codice di errore, l'evento della flotta 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 GameLift servizio Amazon 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.

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 dei metodi di callback, implementati nel codice del server di gioco, che il GameLift servizio Amazon 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 che Amazon GameLift acquisisca e memorizzi.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto 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 GameLift servizio Amazon che il processo del server è pronto per ospitare 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 può 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, che il GameLift servizio Amazon 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 che Amazon GameLift acquisisca e memorizzi.

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;
}
```

RemovePlayerSession()

Notifica al GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift modifica lo slot del giocatore rendendolo disponibile, il che consente di assegnarlo a un nuovo giocatore.

Sintassi

```
GenericOutcome RemovePlayerSession(
    const std::string& playerId);
```

Parametri

playerSessionId

ID univoco emesso dal GameLift servizio Amazon in risposta a una chiamata all'azione [CreatePlayerSession](#) dell'GameLiftAPI Amazon 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 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 FlexMatch riempimento](#).

Questa operazione è asincrona. Se i nuovi giocatori vengono abbinati con successo, il GameLift servizio Amazon fornisce dati aggiornati sul 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. Queste informazioni sono facoltative; se non viene fornito alcun ID, Amazon GameLift 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

Restituisce un `StartMatchBackfillOutcome` oggetto con il ticket di backfill corrispondente o un errore con un messaggio di errore. Lo stato del ticket può essere monitorato utilizzando l'azione AWS SDK [DescribeMatchmaking\(\)](#).

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 SDK [StopMatchmaking\(\)](#). Scopri di più sulla [funzione di FlexMatch riempimento](#).

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 GameLift servizio Amazon che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimarrà attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere richiamato solo dopo il completamento della procedura di chiusura della sessione di gioco, poiché segnala ad Amazon GameLift che il processo del server è immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene chiamata se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Chiama invece [ProcessEnding\(\)](#) per segnalare che sia la sessione di gioco che il processo del 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 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.

```
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. Usa questo metodo dopo aver terminato un processo del server di gioco per evitare di sprecare la memoria del server.

Sintassi

```
GenericOutcome Aws::GameLift::Server::Destroy();
```

Parametri

Non ci sono parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

Esempio

Questo esempio pulisce la memoria allocata da `initSDK` al termine di un processo del server di gioco.

```
if (Aws::GameLift::Server::ProcessEnding().IsSuccess()) {  
    Aws::GameLift::Server::Destroy();  
    exit(0);  
}
```

Riferimento Amazon GameLift Server SDK (C++): tipi di dati

Puoi usare questo riferimento all'SDK per server Amazon GameLift C++ per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

L'API è definita in `GameLiftServerAPI.h`, `LogParameters.h` e `ProcessParameters.h`.

- [Operazioni](#)
- Tipi di dati

DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Puoi utilizzarlo come segue:

- Fornisci un `PlayerSessionId` modulo per richiedere una sessione specifica per il giocatore.
- Fornisci un `GameSessionId` per richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata.
- Fornisci un `PlayerId` 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 `NextToken` per ottenere risultati sotto forma di un insieme 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. Gli ID giocatore sono definiti dallo sviluppatore. Consultare [Genera gli ID dei giocatori](#).

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 che Amazon GameLift carichi e archivi una volta terminata la sessione di gioco. Queste informazioni vengono comunicate al GameLift servizio Amazon durante una [ProcessReady\(\)](#) chiamata.

Indice

logPaths

Percorsi di directory verso i file di registro del server di gioco che desideri che Amazon GameLift memorizzi 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 dei log 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 il set di parametri inviati al GameLift servizio Amazon durante 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

onStartGameSession

Nome della funzione di callback che il GameLift servizio Amazon richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta alla richiesta del cliente [CreateGameSession](#). La funzione di callback passa un [GameSession](#) oggetto (definito nell'Amazon GameLift Service API Reference).

Tipo: `const std::function<void(Aws::GameLift::Model::GameSession)>`
`onStartGameSession`

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che il GameLift servizio Amazon richiama per forzare l'arresto del processo del server. Dopo aver richiamato questa funzione, Amazon GameLift attende cinque minuti che il processo del server si interrompa 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 il GameLift servizio Amazon richiama per richiedere un rapporto sullo stato di salute dal processo del server. Amazon GameLift chiama questa funzione ogni 60 secondi. Dopo aver richiamato questa funzione, Amazon GameLift attende 60 secondi per ricevere una risposta e, se non ne riceve alcuna, registra il processo del server come non integro.

Tipo: `std::function<bool()>` `onHealthCheck`

Campo obbligatorio: no

onUpdateGameSession

Nome della funzione di callback che il GameLift servizio Amazon richiama per passare un oggetto della sessione di gioco aggiornato al processo del server. Amazon GameLift richiama questa

funzione quando viene elaborata una richiesta di [backfill della partita](#) per fornire dati aggiornati sul matchmaker. Passa un [GameSession](#) oggetto, uno status update (updateReason) e l'ID del ticket di backfill corrispondente.

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 GameLift servizio Amazon durante una [StartMatchBackfill\(\)](#) chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco. L'operazione API [GetGameSessionId\(\)](#) restituisce l'identificatore nel formato ARN.

Tipo: String

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 del matchmaker in [Word con i dati del matchmaker](#).

Tipo: String

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. Consulta la Amazon GameLift API Reference Guide per una descrizione del formato degli oggetti di Player. Per trovare gli attributi dei giocatori, gli ID

e gli incarichi delle squadre, osservare l'oggetto delle sessioni di gioco nella proprietà dei dati del matchmaker. Se la latenza viene utilizzata dal matchmaker, raccogliere la latenza aggiornata per la regione attuale e includerla nei dati di ciascun giocatore.

Tipo: `std::vector` https://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 ne GameLift 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 GameLift servizio Amazon durante una [StopMatchBackfill\(\)](#) chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco associato alla richiesta in fase di annullamento.

Tipo: `String`

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco del matchmaker a cui è stata inviata questa richiesta.

Tipo: `String`

Campo obbligatorio: sì

TicketId

Identificatore univoco del ticket di richiesta di backfill da annullare.

Tipo: String

Campo obbligatorio: sì

Riferimento SDK per GameLift server Amazon per C#

Puoi usare questo riferimento all'SDK per server Amazon GameLift C# per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)
- [Riferimento Amazon GameLift Server SDK 4.x per C#](#)

Riferimento Amazon GameLift Server SDK 5.x per C# e Unity

Puoi usare questo riferimento all'SDK 5.x per server Amazon GameLift C# per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, vedere [Aggiungi Amazon GameLift al tuo server di gioco](#) e per informazioni sull'utilizzo del plug-in SDK del server C# per Unity, vedere. [Integra Amazon GameLift in un progetto Unity](#) L'Amazon GameLift Server SDK 5.x per C# supporta .NET 4.6 e .NET 6.

Argomenti

- [Riferimento all'SDK GameLift del server Amazon per C# e Unity: azioni](#)
- [Riferimento SDK per GameLift server Amazon per C# e Unity: tipi di dati](#)

Riferimento all'SDK GameLift del server Amazon per C# e Unity: azioni

Questo riferimento all'SDK del server Amazon GameLift C# ti aiuta a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#) e per informazioni sull'utilizzo del plug-in SDK del server C# per Unity, consulta. [Integra Amazon GameLift in un progetto Unity](#)

Azioni

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetTerminationTime\(\)](#)
- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)
- [Distruigi \(\)](#)

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 SDK per una flotta EC2 gestita. Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo legge i parametri del server dall'ambiente host per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
GenericOutcome InitSDK();
```

Valore restituito

In caso di successo, restituisce un `InitSdkOutcome` oggetto per indicare che il processo del server è pronto per la chiamata [ProcessReady\(\)](#).

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 SDK per una flotta. Anywhere Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo richiede parametri espliciti del server per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
GenericOutcome InitSDK(ServerParameters serverParameters);
```

Parametri

[ServerParameters](#)

Per inizializzare un server di gioco su una GameLift Anywhere flotta Amazon, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket usato per connetterti al tuo 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 GameLift flotta Amazon contenente i tuoi dati di GameLift Anywhere calcolo Amazon.
- Il token di autorizzazione generato dall' GameLift operazione Amazon.

Valore restituito

In caso di successo, restituisce un `InitSdkOutcome` oggetto per indicare che 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

```
//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 ad Amazon GameLift 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 ad Amazon GameLift che il processo del server sta terminando. Richiama 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 ad Amazon GameLift che il processo del server ha attivato una sessione di gioco ed è ora pronto per ricevere le connessioni dei giocatori. Questa azione deve essere richiamata come parte della funzione di `onStartGameSession()` 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 `onStartGameSession()`.

```
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 da Amazon GameLift. Amazon GameLift chiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto ad Amazon GameLift.
- Quando si interrompe 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 ad Amazon GameLift che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID di sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerId)
```

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 gestire una richiesta di connessione, inclusi la convalida e il rifiuto di ID di sessioni giocatore non validi.

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerId)  
{
```

```
GenericOutcome acceptPlayerSessionOutcome =
GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
if(acceptPlayerSessionOutcome.Success)
{
    connectionToSessionMap.emplace(connection, playerSessionId);
    connection.Accept();
}
else
{
    connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);
}
}
```

RemovePlayerSession()

Notifica ad Amazon GameLift che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift cambia lo slot del giocatore rendendolo disponibile.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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

```
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 NextToken e impostando il valore Limite su 10, Amazon GameLift restituirà i primi 10 record delle sessioni 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 fornisce dati aggiornati sui 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 [FlexMatch riempimento](#).

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 GameLift Anywhere flotta Amazon. Per ulteriori informazioni, consulta, [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 GameLift gestita da Amazon, 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 GameLift Amazon 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 GameLift gioco Amazon dalla memoria. Come best practice, richiama 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 reinizializza prima di notificare `InitSDK()` ad Amazon GameLift che il processo è pronto per ospitare una sessione di gioco con. `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);
}
```

Riferimento SDK per GameLift server Amazon per C# e Unity: tipi di dati

Questa guida di riferimento all'SDK per server Amazon GameLift C# può aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, vedere [Aggiungi Amazon GameLift al tuo server di gioco](#) e per informazioni sull'utilizzo del plug-in SDK del server C# per Unity, vedere. [Integra Amazon GameLift in un progetto Unity](#)

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 quali file generati durante una sessione di gioco desideri che il server di gioco carichi su Amazon al GameLift termine della sessione di gioco. Il server di gioco comunica con `LogParameters` to Amazon GameLift durante una [ProcessReady\(\)](#) chiamata.

Proprietà	Descrizione
<code>LogPaths</code>	L'elenco dei percorsi delle directory dei file di registro del server di gioco che desideri che Amazon GameLift memorizzi 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 principale di creazione del gioco.

I percorsi dei log 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 troverà `c:\game\MyGame\sessionLogs` su un'istanza di Windows.

Tipo: `List<String>`

Required: No

ProcessParameters

Questo tipo di dati contiene il set di parametri inviati GameLift ad Amazon durante una [ProcessReady\(\)](#) chiamata.

Proprietà	Descrizione
LogParameters	<p>L'oggetto con un elenco di percorsi di directory per i file di registro delle sessioni di gioco.</p> <p>Tipo: <code>Aws::GameLift::Server::LogParameters</code></p> <p>Required: Yes</p>
OnHealthCheck	<p>Il nome della funzione di callback che Amazon GameLift richiama per richiedere un rapporto sullo stato di salute dal processo del server. Amazon GameLift chiama questa funzione ogni 60 secondi. Dopo aver richiamato questa funzione, Amazon GameLift attende 60 secondi per ricevere una risposta, se non ne viene ricevuta alcuna, Amazon GameLift registra il processo del server come non integro.</p> <p>Tipo: <code>void OnHealthCheckDelegate()</code></p>

Required: Yes

OnProcessTerminate

Il nome della funzione di callback che Amazon GameLift richiama per forzare l'arresto del processo del server. Dopo aver richiamato questa funzione, Amazon GameLift attende cinque minuti che il processo del server si spenga e risponda con una [ProcessEnding\(\)](#) chiamata prima di chiudere il processo del server.

Tipo: void OnProcessTerminate
Delegate()

Required: Yes

OnStartGameSession

Il nome della funzione di callback che Amazon GameLift richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta alla richiesta del cliente [CreateGameSession](#). La funzione di callback accetta un [GameSession](#) oggetto definito nell'Amazon GameLift API Reference.

Tipo: void OnStartGameSession
Delegate([GameSession](#))

Required: Yes

OnUpdateGameSession

Il nome della funzione di callback che Amazon GameLift richiama per passare un oggetto della sessione di gioco aggiornato al processo del server. Amazon GameLift chiama questa funzione quando viene elaborata una richiesta di backfill della partita per fornire dati aggiornati sul matchmaker. Passa un [GameSession](#) oggetto, un aggiornamento di stato (updateReason) e l'ID del ticket di backfill corrispondente.

Tipo: void OnUpdateGameSessionDelegate ()
[UpdateGameSession](#)

Required: No

Porta

Il numero di porta che il processo del server ascolta per le connessioni dei 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

Required: Yes

UpdateGameSession

Le informazioni aggiornate per 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 backfill, questo tipo di dati include l'ID del ticket di backfill.

Proprietà	Descrizione
GameSession	<p>Un GameSession oggetto definito dall'Game LiftAPI Amazon. L'GameSession oggetto contiene proprietà che descrivono una sessione di gioco.</p> <p>Tipo: GameSession GameSession()</p> <p>Required: Yes</p>
UpdateReason	<p>Il motivo per cui la sessione di gioco viene aggiornata.</p> <p>Tipo: UpdateReason UpdateReason()</p> <p>Required: Yes</p>
BackfillTicketId	<p>L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco.</p> <p>Tipo: String</p> <p>Required: Yes</p>

GameSession

Dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	<p>Un identificatore univoco per la sessione di gioco. Una sessione di gioco ARN ha il seguente formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> .</p> <p>Tipo: String</p>

Proprietà	Descrizione
	Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: <code>String</code> Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco. Tipo: <code>String</code> Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: <code>Integer</code> Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: <code>Integer</code> Required: No
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: <code>String</code> Required: No

Proprietà	Descrizione
GameSessionData	<p>Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
MatchmakerData	<p>Le informazioni sul processo di matchmaking utilizzate per creare la sessione di gioco, in sintassi JSON, formattate come una stringa. Oltre alla configurazione di matchmaking utilizzata, contiene dati su tutti i giocatori assegnati alla partita, inclusi gli attributi dei giocatori e le assegnazioni delle squadre.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
GameProperties	<p>Un set 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.</p> <p>Tipo: <code>Dictionary<string, string></code></p> <p>Required: No</p>

Proprietà	Descrizione
DnsName	<p>L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato:</p> <ul style="list-style-type: none"> • Flotte abilitate a TLS: <code><unique identifier>.<region identifier>.amazon.gamelift.com</code> • Flotte non abilitate a TLS: <code>ec2-<unique identifier>.compute.amazonaws.com</code> <p>Quando ti connetti a una sessione di gioco in esecuzione su un parco veicoli abilitato a TLS, devi utilizzare il nome DNS, non l'indirizzo IP.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

ServerParameters

Informazioni utilizzate per mantenere la connessione tra un GameLift Anywhere server Amazon e il GameLift servizio Amazon. Queste informazioni vengono utilizzate quando si avviano nuovi processi server con [InitSDK\(\)](#). Per i server ospitati su istanze EC2 GameLift gestite da Amazon, utilizza un oggetto vuoto.

Proprietà	Descrizione
WebSocketUrl	<p>Sono <code>GameLiftServerSdkEndpoint</code> stati restituiti <code>RegisterCompute</code> quando fai parte di Amazon GameLiftAnywhere.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>

Proprietà	Descrizione
ProcessId	<p>Un identificatore univoco registrato nel processo server che ospita il gioco.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>
HostId	<p>Un identificatore univoco per l'host con i processi del server che ospita il gioco. L'hostID viene <code>ComputeName</code> utilizzato quando hai registrato il tuo computer. Per ulteriori informazioni, vedere RegisterCompute</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>
FleetId	<p>L'ID della flotta su cui è registrato il computer. Per ulteriori informazioni, vedere, RegisterCompute.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>
AuthToken	<p>Il token di autenticazione generato da Amazon GameLift che autentica il tuo server su Amazon. GameLift Per ulteriori informazioni, vedere, GetComputeAuthToken.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di backfill di matchmaking. Il server di gioco comunica queste informazioni ad Amazon durante GameLift una [StartMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>L'identificatore univoco della sessione di gioco. L'operazione API GetGameSessionId restituisce l'identificatore in formato ARN.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>
MatchmakingConfigurationArn	<p>L'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 dei dati del matchmaker. Scopri di più sui dati del matchmaker in Lavora con i dati del matchmaker.</p> <p>Tipo: <code>String</code></p> <p>Required: Yes</p>
Players	<p>Un set di dati che rappresenta tutti i giocatori che sono attualmente nella sessione di gioco. Il matchmaker utilizza queste informazioni per cercare nuovi giocatori che rappresentano un buon abbinamento per i giocatori attuali.</p> <p>Tipo: <code>List<Player></code></p> <p>Required: Yes</p>
TicketId	<p>L'identificatore univoco per un ticket di richiesta di matchmaking o match backfill. Se non fornisci un valore, Amazon ne GameLift genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill</p>

Proprietà	Descrizione
	<p>degli abbinamenti o annullare la richiesta, se necessario.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

Player

Rappresenta un giocatore in matchmaking. Quando inizia una richiesta di matchmaking, un giocatore ha un ID giocatore, attributi ed eventualmente dati di latenza. Amazon GameLift aggiunge le informazioni sulla squadra dopo la partita.

Proprietà	Descrizione
LatencyInMS	<p>Un insieme di valori espressi in millisecondi, che indicano la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione.</p> <p>Se viene utilizzata questa proprietà, il giocatore viene abbinato solo alle posizioni elencate. Se un matchmaker ha una regola che valuta la latenza dei giocatori, i giocatori devono segnalare la latenza per essere abbinati.</p> <p>Tipo: <code>Dictionary<string, int></code></p> <p>Required: No</p>
PlayerAttributes	<p>Una raccolta di coppie chiave:valore che contengono informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi dei giocatori devono corrispondere a quelle <code>PlayerAttributes</code> utilizzate in un set di regole di matchmaking.</p>

Proprietà	Descrizione
	<p>Per ulteriori informazioni sugli attributi dei giocatori, consulta AttributeValue.</p> <p>Tipo: Dictionary<string, Attribute Value</p> <p>Required: No</p>
PlayerId	<p>Un identificatore univoco per un giocatore.</p> <p>Tipo: String</p> <p>Required: No</p>
Team	<p>Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking.</p> <p>Tipo: String</p> <p>Required: No</p>

DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Può essere usato in diversi modi: (1) fornire PlayerSessionId a richiedere una sessione di gioco specifica; (2) fornire GameSessionId a richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata; o (3) fornire PlayerId a per richiedere tutte le sessioni dei giocatori 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	<p>L'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</p>

Proprietà	Descrizione
	<p>sessione di gioco è il seguente: <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string></code> . Il valore di <code><ID string></code> 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.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
<code>PlayerSessionId</code>	<p>L'identificatore univoco per la sessione di un giocatore.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
<code>PlayerId</code>	<p>L'identificatore univoco di un giocatore. Consultare Genera gli ID dei giocatori.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

Proprietà	Descrizione
PlayerSessionStatusFilter	<p>Lo stato della sessione del giocatore su cui filtrare i risultati. Tra gli stati sessione giocatore possibili sono inclusi i seguenti:</p> <ul style="list-style-type: none">• 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). <p>Tipo: <code>String</code></p> <p>Required: No</p>
NextToken	<p>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.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

Proprietà	Descrizione
Limite	<p>Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di backfill di matchmaking. Il server di gioco comunica queste informazioni al GameLift servizio Amazon durante una [StopMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>L'identificatore univoco della sessione di gioco della richiesta di annullamento.</p> <p>Tipo: <code>string</code></p> <p>Required: Yes</p>
MatchmakingConfigurationArn	<p>L'identificatore univoco del matchmaker a cui è stata inviata questa richiesta.</p> <p>Tipo: <code>string</code></p> <p>Required: Yes</p>
TicketId	<p>L'identificatore univoco del ticket di richiesta di backfill da annullare.</p> <p>Tipo: <code>string</code></p> <p>Required: Yes</p>

GetFleetRoleCredentialsRequest

Questo tipo di dati offre al server di gioco un accesso limitato alle tue altre AWS risorse. Per ulteriori informazioni, consultare [Configura un ruolo di servizio IAM per Amazon GameLift](#).

Proprietà	Descrizione
RoleArn	Amazon Resource Name (ARN) del ruolo di servizio che estende l'accesso limitato alle tue AWS risorse. Tipo: <code>string</code> Required: Yes
RoleSessionName	Il nome della sessione che descrive l'uso delle credenziali del ruolo. Tipo: <code>string</code> Required: No

AttributeValue

Usa questi valori nelle coppie chiave-valore [Player](#) degli attributi. Questo oggetto consente di specificare un valore di attributo utilizzando uno qualsiasi dei tipi di dati validi: stringa, numero, array di stringhe o mappa dati. Ogni `AttributeValue` oggetto può utilizzare solo una delle proprietà disponibili.

Proprietà	Descrizione
Tipo ATR	Specifica il tipo di valore dell'attributo. Tipo: un valore <code>AttrType</code> enum . Required: No
S	Rappresenta un valore di attributo stringa. Tipo: <code>string</code>

Proprietà	Descrizione
	Required: Yes
N	Rappresenta un valore numerico di attributo. Tipo: <code>double</code> Required: Yes
SL	Rappresenta una matrice di valori di attributi di stringa. Tipo: <code>string[]</code> Required: Yes
SDM	Rappresenta un dizionario di chiavi di stringa e valori doppi. Tipo: <code>Dictionary<string, double></code> Required: Yes

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: <code>string</code> Required: No
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> Required: Yes

Proprietà	Descrizione
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

GenericOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Success (Riuscito)	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Required: Yes</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “GameLiftError”</p> <p>Required: No</p>

DescribePlayerSessionsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “DescribePlayerSessionsResult”</p>

Proprietà	Descrizione
	Required: No
Success (Riuscito)	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Required: Yes</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

DescribePlayerSessionsResult

Proprietà	Descrizione
NextToken	<p>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.</p> <p>Tipo: <code>string</code></p> <p>Required: Yes</p>
PlayerSessions	<p>Una raccolta di oggetti contenenti proprietà per ogni sessione del giocatore che corrisponde alla richiesta.</p> <p>Tipo: <code>IList<the section called "PlayerSession"></code></p> <p>Richiesto:</p>

Proprietà	Descrizione
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: bool Required: Yes
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called "GameLiftError" Required: No

PlayerSession

Proprietà	Descrizione
CreationTime	Tipo: long Required: Yes
FleetId	Tipo: string Required: Yes
GameSessionId	Tipo: string Required: Yes
IpAddress	Tipo: string Required: Yes
PlayerData	Tipo: string Required: Yes
PlayerId	Tipo: string

Proprietà	Descrizione
	Required: Yes
PlayerSessionId	Tipo: <code>string</code> Required: Yes
Porta	Tipo: <code>int</code> Required: Yes
Stato	Tipo: A <code>PlayerSessionStatus</code> enum. Required: Yes
TerminationTime	Tipo: <code>long</code> Required: Yes
DnsName	Tipo: <code>string</code> Required: Yes

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 “StartMatchBackfill IResult” Required: No
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code>

Proprietà	Descrizione
	Required: Yes
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 Required: Yes

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
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: bool Required: Yes

Proprietà	Descrizione
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

GetComputeCertificateResult

Il percorso del certificato TLS sul tuo computer e il nome host del computer.

Proprietà	Descrizione
CertificatePath	<p>Tipo: string</p> <p>Required: Yes</p>
ComputeName	<p>Tipo: string</p> <p>Required: Yes</p>

GetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called "GetFleetRoleCredentialsResult"</p> <p>Required: No</p>
Success (Riuscito)	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p>

Proprietà	Descrizione
	Required: Yes
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 AWS risorse. Tipo: <code>string</code> Required: No
AssumedRoleId	L'ID dell'utente a cui appartiene il ruolo del servizio. Tipo: <code>string</code> Required: No
AssumedRoleUserArn	Amazon Resource Name (ARN) dell'utente a cui appartiene il ruolo di servizio. Tipo: <code>string</code> Required: No
Expiration	Il periodo di tempo fino alla scadenza delle credenziali di sessione. Tipo: <code>DateTime</code>

Proprietà	Descrizione
	Required: No
SecretAccessKey	L'ID della chiave di accesso segreta per l'autenticazione. Tipo: <code>string</code> Required: No
SessionToken	Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse. Tipo: <code>string</code> Required: No
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> Required: Yes
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: <code>DateTime</code>

Proprietà	Descrizione
	Required: No
Success (Riuscito)	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> Required: Yes
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: A <code>GameLiftErrorType</code> enum . Required: No
ErrorMessage	Messaggio di errore. Tipo: <code>string</code> Required: No
ErrorName	Il nome dell'errore. Tipo: <code>string</code> Required: No

enumerazioni;

Le enumerazioni definite per Amazon GameLift Server SDK (C#) sono definite come segue:

AttrType

- NONE
- CORDA
- DOPPIO
- LISTA DI 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 ad Amazon non è riuscita. GameLift
- NETWORK_NOT_INITIALIZED — La rete non è stata inizializzata.
- GAMESESSION_ID_NOT_SET — L'ID della sessione di gioco non è stato impostato.
- ECCEZIONE_RICHIESTA ERRATA
- ECCEZIONE_SERVIZIO INTERNO
- ALREADY_INITIALIZED: il GameLift server o il client Amazon è già stato inizializzato con Initialize ().
- FLEET_MISMATCH: la flotta di destinazione non corrisponde alla flotta di una GameSession o PlayerSession.
- GAMELIFT_CLIENT_NOT_INITIALIZED — Il client Amazon non è stato inizializzato. GameLift
- GAMELIFT_SERVER_NOT_INITIALIZED — Il server Amazon non è stato inizializzato. GameLift
- GAME_SESSION_ENDED_FAILED: Amazon GameLift Server SDK non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco.
- GAME_SESSION_NOT_READY — La sessione di gioco Amazon GameLift Server non è stata attivata.
- GAME_SESSION_READY_FAILED: Amazon GameLift Server SDK non è riuscito a contattare il servizio per segnalare che la sessione di gioco è pronta.

- `INITIALIZATION_MISMATCH` — Un metodo client è stato chiamato dopo `Server: :Initialize ()` o viceversa.
- `NOT_INITIALIZED`: il GameLift server o il client Amazon non è stato inizializzato 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 destinazione.
- `PROCESS_ENDING_FAILED`: Amazon GameLift Server SDK non è riuscito a contattare il servizio per segnalare la fine del processo.
- `PROCESS_NOT_ACTIVE` — Il processo del server non è ancora attivo, non è associato a un `GameSession` e non può essere accettato o elaborato. `PlayerSessions`
- `PROCESS_NOT_READY` — Il processo del server non è ancora pronto per essere attivato.
- `PROCESS_READY_FAILED`: Amazon GameLift Server SDK non è riuscito a contattare il servizio per 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 `xStX` non è riuscita.
- `STX_INITIALIZATION_FAILED` — L'inizializzazione del componente di backend del server `xStX` non è stata eseguita.
- `UNEXPECTED_PLAYER_SESSION` — Il server ha riscontrato una sessione di giocatore non registrata.
- `ERRORE WEBSOCKET_CONNECT_DEFAULT`
- `ERRORE DI CONNESSIONE WEB PROIBITO`
- `WEBSOCKET_CONNECT_FAILURE_URL NON VALIDO`
- `TIMEOUT DI INTERRUZIONE DELLA CONNESSIONE WEB`
- `WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE` — Errore recuperabile nell'invio di un messaggio al Servizio. `GameLift WebSocket`
- `WEBSOCKET_SEND_MESSAGE_FAILURE` — Mancato invio di un messaggio al Servizio. `GameLift 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 le sessioni dei nuovi giocatori.

PlayerSessionStatus

- ATTIVO
- COMPLETED
- NON IMPOSTATO
- RISERVATO
- SCADUTO

Riferimento Amazon GameLift Server SDK 4.x per C#

Questa guida di riferimento per Amazon GameLift C# Server SDK 4.x può aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento Amazon GameLift Server SDK \(C#\): azioni](#)
- [Riferimento Amazon GameLift Server SDK \(C#\): tipi di dati](#)

Riferimento Amazon GameLift Server SDK (C#): azioni

Puoi usare questo riferimento all'SDK per server Amazon GameLift C# per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

- Operazioni
- [Tipi di dati](#)

AcceptPlayerSession()

Notifica al GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID della sessione

del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot per il giocatore nella sessione di gioco. Una volta convalidato, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO a ATTIVO.

Sintassi

```
GenericOutcome AcceptPlayerSession(String playerId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon GameLift quando viene creata una nuova sessione giocatore. Un ID di sessione del giocatore è specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLiftAPI [StartGameSessionPlacementCreateGameSession](#), [DescribeGameSessionPlacement](#), o [DescribePlayerSessions](#).

Tipo: String

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 gestire una richiesta di connessione, inclusi la convalida e il rifiuto di ID di sessioni giocatore non validi.

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerId){
    var acceptPlayerSessionOutcome =
    GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
    if(acceptPlayerSessionOutcome.Success)
    {
        connectionToSessionMap.emplace(connection, playerId);
        connection.Accept();
    }
    else
```

```
{
    connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);
}
```

ActivateGameSession()

Notifica al GameLift servizio Amazon che il processo del server ha attivato una sessione di gioco ed è ora 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 [PlayerSession](#) dati dell'GameLiftAPI Amazon AWS SDK.

Esempio

Questo esempio illustra una richiesta per tutte le sessioni giocatore attivamente connesse a una sessione di gioco specificata. Omettendo `NextToken` impostando il valore limite su 10, Amazon GameLift restituirà i record delle sessioni dei primi 10 giocatori 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 al parco veicoli e alle relative istanze. AWS Certificate Manager genera questo certificato quando si crea una nuova flotta 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 certificati TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato principale, 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 [GetCertificateResponse Elements](#) in the 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 del server esegue questa azione dopo aver ricevuto una `onProcessTerminate()` richiamata dal GameLift servizio Amazon. [Amazon GameLift può](#)

[chiamare `onProcessTerminate\(\)` per i seguenti motivi: \(1\) per problemi di salute \(il processo del server ha segnalato l'integrità delle porte o non ha risposto ad Amazon\)GameLift, \(2\) quando si interrompe l'istanza durante un evento di ridimensionamento o \(3\) quando un'istanza viene interrotta a causa di un'interruzione di un'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 `AwsDateTimeOutcome` oggetto. Il valore è il tempo di terminazione, espresso in zecche trascorse da 0001 00:00:00. Ad esempio, il valore data/ora 2020-09-13 12:26:40 -000Z è uguale a 637355968000000000 zecche. Se non è disponibile alcun orario di terminazione, restituisce un messaggio di errore.

Esempio

```
var getTerminationTimeOutcome = GameLiftServerAPI.GetTerminationTime();
```

InitSDK()

Inizializza l'Amazon GameLift SDK. Questo metodo deve essere richiamato all'avvio, prima che si verifichi qualsiasi altra inizializzazione GameLift relativa ad Amazon.

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 chiamata [ProcessReady\(\)](#).

Esempio

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

ProcessEnding()

Notifica al GameLift servizio Amazon che il processo del server si sta spegnendo. 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 che il metodo esce con un codice 0, puoi terminare il processo con un codice di uscita riuscito. Puoi anche uscire dal processo con un codice di errore. Se esci con un codice di errore, l'evento della flotta 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 GameLift servizio Amazon 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

```
GenericOutcome ProcessReady(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, che il GameLift servizio Amazon 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 che Amazon GameLift acquisisca e memorizzi.

Campo obbligatorio: sì

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
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 GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift modifica lo slot del giocatore rendendolo disponibile, il che consente di assegnarlo a un nuovo giocatore.

Sintassi

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon GameLift quando viene creata una nuova sessione giocatore. Un ID di sessione del giocatore è specificato in un `PlayerSession` oggetto, che viene restituito in risposta a una chiamata del client alle azioni GameLiftAPI [StartGameSessionPlacementCreateGameSession](#), [DescribeGameSessionPlacement](#), o [DescribePlayerSessions](#).

Tipo: String

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 FlexMatch riempimento](#).

Questa operazione è asincrona. Se i nuovi giocatori vengono abbinati con successo, il GameLift servizio Amazon 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 oggetto [StartMatchBackfillRequest](#) 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 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

Restituisce un `StartMatchBackfillOutcome` oggetto con l'ID del ticket di backfill corrispondente o 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 FlexMatch riempimento](#).

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 GameLift servizio Amazon che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimarrà attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere richiamato solo dopo il completamento della procedura di chiusura della sessione di gioco, poiché segnala ad Amazon GameLift che il processo del server è immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene chiamata se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Chiama invece [ProcessEnding\(\)](#) per segnalare che sia la sessione di gioco che il processo del 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\(\)](#) nell'Amazon GameLift Service 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 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.

```
var updatePlayerSessionCreationPolicyOutcomex =  
  
GameLiftServerAPI.UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy.ACCEPT_ALL);
```

Riferimento Amazon GameLift Server SDK (C#): tipi di dati

Puoi usare questo riferimento all'SDK per server Amazon GameLift C# per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

- [Operazioni](#)
- Tipi di dati

LogParameters

Questo tipo di dati viene utilizzato per identificare i file generati durante una sessione di gioco che desideri che Amazon GameLift carichi e archivi una volta terminata la sessione di gioco. Queste informazioni vengono comunicate al GameLift servizio Amazon durante una [ProcessReady\(\)](#) chiamata.

Indice

logPaths

Elenco dei percorsi delle directory dei file di log del server di gioco che desideri che Amazon GameLift memorizzi 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 dei log 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 usato in diversi modi: (1) fornire PlayerSessionId a richiedere una sessione di gioco specifica; (2) fornire GameSessionId a richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata; o (3) fornire PlayerId a per richiedere tutte le sessioni dei giocatori 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 `NextToken` per ottenere risultati sotto forma di un insieme 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. Gli ID giocatore sono definiti dallo sviluppatore. Consultare [Genera gli ID dei giocatori](#).

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 il set di parametri inviati al GameLift servizio Amazon durante 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 il GameLift servizio Amazon richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta alla richiesta del cliente [CreateGameSession](#). La funzione di callback accetta un [GameSession](#) oggetto (definito nell'Amazon GameLift Service API Reference).

Tipo: `void OnStartGameSessionDelegate(GameSession gameSession)`

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che il GameLift servizio Amazon richiama per forzare l'arresto del processo del server. Dopo aver richiamato questa funzione, Amazon GameLift attende cinque minuti che il processo del server si spenga e risponda con una [ProcessEnding\(\)](#) chiamata prima di chiudere il processo del server.

Tipo: `void OnProcessTerminateDelegate()`

Campo obbligatorio: sì

onHealthCheck

Nome della funzione di callback che il GameLift servizio Amazon richiama per richiedere un rapporto sullo stato di salute dal processo del server. Amazon GameLift chiama questa funzione ogni 60 secondi. Dopo aver richiamato questa funzione, Amazon GameLift attende 60 secondi per ricevere una risposta e, se non ne riceve alcuna, registra il processo del server come non integro.

Tipo: `bool OnHealthCheckDelegate()`

Campo obbligatorio: sì

onUpdateGameSessione

Nome della funzione di callback che il GameLift servizio Amazon richiama per passare un oggetto della sessione di gioco aggiornato al processo del server. Amazon GameLift richiama questa

funzione quando viene elaborata una richiesta di [backfill della partita](#) per fornire dati aggiornati sul matchmaker. Passa un [GameSession](#) oggetto, un aggiornamento di stato (updateReason) e l'ID del ticket di backfill corrispondente.

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 GameLift servizio Amazon durante una [StartMatchBackfill\(\)](#) chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco. Il metodo SDK [GetGameSessionId\(\)](#) restituisce l'identificatore in formato ARN.

Tipo: String

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 del matchmaker in [Lavora con i dati del matchmaker](#).

Tipo: String

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. Consulta la Amazon GameLift API Reference Guide per una descrizione del formato degli oggetti di Player. Per trovare gli attributi dei giocatori, gli ID e gli incarichi delle squadre, osservare l'oggetto delle sessioni di gioco nella proprietà dei dati del

matchmaker. 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 ne GameLift 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 GameLift servizio Amazon durante una [StopMatchBackfill\(\)](#) chiamata.

Indice

GameSessionArn

Identificatore univoco della sessione di gioco associato alla richiesta in fase di annullamento.

Tipo: String

Campo obbligatorio: sì

MatchmakingConfigurationArn

Identificatore univoco del matchmaker a cui è stata inviata questa richiesta.

Tipo: String

Campo obbligatorio: sì

TicketId

Identificatore univoco del ticket di richiesta di backfill da annullare.

Tipo: String

Campo obbligatorio: sì

Riferimento SDK GameLift del server Amazon per Go

Puoi usare questo riferimento all'SDK del server Amazon GameLift Go per aiutarti a preparare il tuo gioco multiplayer da utilizzare con AmazonGameLift. Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento all'SDK \(Go\) per GameLift server Amazon: Azioni](#)
- [Riferimento Amazon GameLift Server SDK \(Go\): tipi di dati](#)

Riferimento all'SDK (Go) per GameLift server Amazon: Azioni

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift Go per aiutarti a preparare il gioco multiplayer da utilizzare con Amazon GameLift. Per dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

`GameLiftServerAPI`.`godefinisce` le azioni dell'SDK del server Go.

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 Amazon GameLift SDK. Chiama questo metodo all'avvio prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo imposta la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
func InitSDK(params ServerParameters) error
```

Parametri

[ServerParameters](#)

Per inizializzare un server di gioco su una GameLift Anywhere flotta Amazon, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket usato per connetterti al tuo 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 GameLift flotta Amazon contenente i tuoi dati di GameLift Anywhere calcolo Amazon.
- Il token di autorizzazione generato dall' GameLift operazione Amazon.

Per inizializzare un server di gioco su una flotta EC2 GameLift gestita da Amazon, crea un `ServerParameters` oggetto senza parametri. Con questa chiamata, l' GameLift agente Amazon configura l'ambiente di elaborazione e si connette automaticamente al GameLift servizio Amazon per te.

Valore restituito

In caso di successo, restituisce un `nil` errore per indicare che 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

GameLift Anywhere Esempio di Amazon

```
//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 GameLift agent to enable
  further communication.
err := server.InitSDK(serverParameters)
```

Esempio di EC2 GameLift gestito da Amazon

```
//Define the server parameters
serverParameters := ServerParameters {}

//Call InitSDK to establish a local connection with the GameLift agent to enable
  further communication.
err := server.InitSDK(serverParameters)
```

ProcessReady()

Notifica ad Amazon GameLift 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 GameLift servizio Amazon per comunicare con il processo del server.
- Il numero di porta su cui è in ascolto il processo del server.
- Il tipo di [LogParameters](#) dati contenente il percorso di tutti i file specifici della sessione di gioco che desideri che Amazon GameLift acquisisca e memorizzi.

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 ad Amazon GameLift che il processo del server sta terminando. Richiama 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 ad Amazon GameLift 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. Amazon GameLift chiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto ad Amazon GameLift.
- Quando si interrompe 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 ad Amazon GameLift che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID di sessione del giocatore sia valido. Dopo la convalida della sessione del giocatore, Amazon GameLift modifica lo stato dello slot del giocatore da `RESERVED` a `ACTIVE`.

Sintassi

```
func AcceptPlayerSession(playerSessionID string) error
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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 gestisce una richiesta di connessione che include la convalida e il rifiuto di ID di sessione giocatore non validi.

```
func ReceiveConnectingPlayerSessionID(conn Connection, playerSessionID string) {
    err := server.AcceptPlayerSession(playerSessionID)
    if err != nil {
        connection.Accept()
    } else {
        connection.Reject(err.Error())
    }
}
```

RemovePlayerSession()

Notifica ad Amazon GameLift che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift cambia lo slot del giocatore rendendolo disponibile.

Sintassi

```
func RemovePlayerSession(playerSessionID string) error
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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

```
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 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 `NextToken` impostando il valore `Limite` su 10, Amazon GameLift restituisce i primi 10 record delle sessioni di gioco 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 [FlexMatch riempimento](#).

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift fornisce dati aggiornati sui 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, Amazon ne GameLift 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 GameLift Anywhere flotta Amazon. 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 GameLift gestita da Amazon, 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 ad altri utenti su Amazon. Servizi AWS GameLift Queste credenziali consentono al server di gioco di utilizzare le tue risorse. AWS Per ulteriori informazioni, consulta [Configura un ruolo di servizio IAM per Amazon GameLift](#).

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 GameLift gioco Amazon dalla memoria. Come best practice, richiama 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 reinizializza prima di notificare `InitSDK()` ad Amazon GameLift che il processo è pronto per ospitare una sessione di gioco con `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)
}
}
```

Riferimento Amazon GameLift Server SDK (Go): tipi di dati

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift Go per aiutarti a preparare il gioco multiplayer da utilizzare con Amazon GameLift. Per dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

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 che Amazon GameLift carichi e archivi al termine della sessione di gioco. Il server di gioco fornisce LogParameters ad Amazon GameLift 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 che Amazon GameLift memorizzi 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. Il processo del server fornisce queste informazioni ad Amazon GameLift con una chiamata a [ProcessReady\(\)](#).

Proprietà	Descrizione
LogParameters	<p>Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift copia e archivia i file per accessi futuri.</p> <p>Tipo: LogParameters</p> <p>Required: No</p>
OnHealthCheck	<p>La funzione di callback GameLift richiamata da Amazon per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift chiama 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 registra il processo del server come non integro.</p> <p>Tipo: <code>OnHealthCheck func() bool</code></p> <p>Required: No</p>
OnProcessTerminate	<p>La funzione di callback GameLift richiamata da Amazon per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift</p>

	<p>attende 5 minuti che il processo server si spenga e risponda con una ProcessEnding() chiamata prima di chiudere il processo server.</p> <p>Tipo: <code>OnProcessTerminate func()</code></p> <p>Campo obbligatorio: sì</p>
<p><code>OnStartGameSession</code></p>	<p>La funzione di callback GameLift richiamata da Amazon per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift chiama questa funzione quando viene elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSession oggetto, uno status update (<code>updateReason</code>) e l'ID del ticket match backfill.</p> <p>Tipo: <code>OnStartGameSession func (model.GameSession)</code></p> <p>Campo obbligatorio: sì</p>
<p><code>OnUpdateGameSession</code></p>	<p>La funzione di callback GameLift richiamata da Amazon per passare informazioni aggiornate sulla sessione di gioco al processo del server. Amazon GameLift chiama questa funzione dopo aver elaborato una richiesta di match backfill per fornire dati aggiornati del matchmaker.</p> <p>Tipo: <code>OnUpdateGameSession func (model.UpdateGameSession)</code></p> <p>Required: No</p>
<p><code>Port</code></p>	<p>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.</p> <p>Tipo: <code>int</code></p> <p>Campo obbligatorio: sì</p>

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	<p>Un GameSession oggetto definito dall' GameLift API Amazon. L'GameSession oggetto contiene proprietà che descrivono una sessione di gioco.</p> <p>Tipo: <code>GameSession</code> <code>GameSession()</code></p> <p>Campo obbligatorio: sì</p>
UpdateReason	<p>Il motivo per cui la sessione di gioco viene aggiornata.</p> <p>Tipo: <code>UpdateReason</code> <code>UpdateReason()</code></p> <p>Campo obbligatorio: sì</p>
BackfillTicketId	<p>L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

GameSession

I dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	<p>Un identificatore univoco per la sessione di gioco. Una sessione di gioco Amazon Resource Name (ARN) ha il seguente formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code></p> <p>Tipo: <code>String</code></p>

Proprietà	Descrizione
	Required: No
Nome	Un'etichetta descrittiva della sessione di gioco. Tipo: <code>String</code> Required: No
FleetId	Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco. Tipo: <code>String</code> Required: No
MaximumPlayerSessionCount	Il numero massimo di connessioni dei giocatori alla sessione di gioco. Tipo: <code>Integer</code> Required: No
Porta	Il numero di porta per la sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: <code>Integer</code> Required: No
IpAddress	L'indirizzo IP della sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta. Tipo: <code>String</code> Required: No

Proprietà	Descrizione
GameSessionData	<p>Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
MatchmakerData	<p>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.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
GameProperties	<p>Un insieme di proprietà personalizzate per una sessione di gioco, formattate e come coppie chiave:valore. Queste proprietà vengono passate con una richiesta di avvio di una nuova sessione di gioco.</p> <p>Tipo: <code>map[string] string</code></p> <p>Required: No</p>

Proprietà	Descrizione
DnsName	<p>L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato:</p> <ul style="list-style-type: none"> • Flotte abilitate per TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> • Flotte non compatibili con TLS: <code>ec2-<unique identifier>.compute.amazonaws.com</code> <p>Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi utilizzare il nome DNS, non l'indirizzo IP.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

ServerParameters

Informazioni utilizzate per mantenere la connessione tra un GameLift Anywhere server Amazon e il GameLift servizio Amazon. Queste informazioni vengono utilizzate quando si avviano nuovi processi server con [InitSDK\(\)](#). Per i server ospitati su istanze EC2 GameLift gestite da Amazon, usa un oggetto vuoto.

Proprietà	Descrizione
WebSocket URL	<p><code>GameLiftServerSdkEndpoint</code> Amazon GameLift restituisce quando RegisterCompute richiedi una risorsa di GameLift Anywhere elaborazione Amazon.</p> <p>Tipo: <code>string</code></p> <p>Campo obbligatorio: sì</p>
ProcessID	<p>Un identificatore univoco registrato nel processo del server che ospita il gioco.</p> <p>Tipo: <code>string</code></p>

Proprietà	Descrizione
	Campo obbligatorio: sì
HostID	<p>L'identificatore univoco della risorsa di elaborazione che ospita il nuovo processo del server.</p> <p>HostIDViene ComputeName utilizzato al momento della registrazione del computer. Per ulteriori informazioni, vedere RegisterCompute.</p> <p>Tipo: string</p> <p>Campo obbligatorio: sì</p>
FleetID	<p>L'identificatore univoco della flotta su cui è registrato il calcolo. Per ulteriori informazioni, vedere. RegisterCompute</p> <p>Tipo: string</p> <p>Campo obbligatorio: sì</p>
AuthToken	<p>Il token di autenticazione generato da Amazon GameLift che autentica il tuo server su Amazon. GameLift Per ulteriori informazioni, consulta. GetComputeAuthToken</p> <p>Tipo: string</p> <p>Campo obbligatorio: sì</p>

StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di riempimento del matchmaking. Il server di gioco comunica queste informazioni ad Amazon GameLift tramite una [StartMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>L'identificatore univoco della sessione di gioco. L'operazione API GetGameSessionId restituisce l'identificatore in formato ARN.</p> <p>Tipo: String</p>

Proprietà	Descrizione
	Campo obbligatorio: sì
MatchmakingConfigurationArn	<p>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.</p> <p>Tipo: <code>String</code></p> <p>Campo obbligatorio: sì</p>
Players	<p>Un insieme 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.</p> <p>Tipo: <code>[]model.Player</code></p> <p>Campo obbligatorio: sì</p>
TicketId	<p>L'identificatore univoco di un ticket di richiesta di matchmaking o match backfill. Se non fornisci un valore, Amazon ne GameLift genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill degli abbinamenti o annullare la richiesta, se necessario.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

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 aggiunge le informazioni sulla squadra dopo la fine di una partita.

Proprietà	Descrizione
LatencyInSM	<p>Un insieme di valori espressi in millisecondi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione.</p> <p>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.</p> <p>Tipo: <code>map[string] int</code></p> <p>Required: No</p>
PlayerAttributes	<p>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 <code>PlayerAttributes</code> utilizzate in un set di regole di matchmaking.</p> <p>Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue.</p> <p>Tipo: <code>map[string] AttributeValue</code></p> <p>Required: No</p>
PlayerId	<p>Un identificatore univoco per un giocatore.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
Team	<p>Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>

DescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo del server fornisce queste informazioni con una [DescribePlayerSessions\(\)](#) chiamata ad Amazon GameLift.

Proprietà	Descrizione
GameSessionID	<p>Un identificatore di sessione di gioco univoco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata.</p> <p>Il formato dell'ID della sessione di gioco è <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string></code>. GameSessionID È una stringa ID personalizzata o una stringa generata.</p> <p>Tipo: String</p> <p>Required: No</p>
PlayerSessionID	<p>L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica.</p> <p>Tipo: String</p> <p>Required: No</p>
PlayerID	<p>L'identificatore univoco di un giocatore. Usa questo parametro per richiedere tutte le sessioni di gioco per un giocatore specifico. Per informazioni, consulta Genera gli ID dei giocatori.</p> <p>Tipo: String</p> <p>Required: No</p>
PlayerSessionStatusFilter	<p>Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono:</p> <ul style="list-style-type: none"> 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.

Proprietà	Descrizione
	<ul style="list-style-type: none"> • 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). <p>Tipo: <code>String</code></p> <p>Required: No</p>
<code>NextToken</code>	<p>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.</p> <p>Tipo: <code>String</code></p> <p>Required: No</p>
<code>Limit</code>	<p>Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>

StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di matchmaking backfill. Il server di gioco comunica queste informazioni al GameLift servizio Amazon tramite una [StopMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
<code>GameSessionArn</code>	<p>L'identificatore univoco della sessione di gioco della richiesta annullata.</p> <p>Tipo: <code>string</code></p> <p>Required: No</p>

Proprietà	Descrizione
MatchmakingConfigurationArn	L'identificatore univoco del matchmaker a cui è stata inviata questa richiesta. Tipo: <code>string</code> Required: No
TicketId	L'identificatore univoco del ticket di richiesta di riempimento da annullare. Tipo: <code>string</code> Required: No

GetFleetRoleCredentialsRequest

Le credenziali del ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco. Per ulteriori informazioni, consulta [Configura un ruolo di servizio IAM per Amazon GameLift](#).

Proprietà	Descrizione
RoleArn	L'ARN del ruolo di servizio che estende l'accesso limitato alle AWS risorse. Tipo: <code>string</code> Campo obbligatorio: sì
RoleSessionName	Il nome della sessione che descrive l'uso delle credenziali del ruolo. Tipo: <code>string</code> Campo obbligatorio: sì

Riferimento SDK GameLift del server Amazon per Unreal Engine

Questo riferimento all'SDK di Amazon GameLift Server può aiutarti a preparare i tuoi progetti di gioco Unreal Engine da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Questa API è definita in `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Per configurare il plugin Unreal Engine e vedere esempi di codice [Integra Amazon GameLift in un progetto Unreal Engine](#).

Argomenti

- [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)
- [Riferimento all'SDK GameLift 3.x del server Amazon Unreal Engine](#)

Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine

Puoi usare questo riferimento all'SDK 5.x del server Amazon GameLift Unreal Engine per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione [Aggiungi Amazon GameLift al tuo server di gioco](#), vedere e per informazioni sull'utilizzo del plug-in del server Unreal SDK, vedere. [Integra Amazon GameLift in un progetto Unreal Engine](#)

Argomenti

- [Riferimento Amazon GameLift Server SDK \(Unreal\) 5.x: azioni](#)
- [Riferimento all'SDK per GameLift server Amazon \(Unreal\): tipi di dati](#)

Riferimento Amazon GameLift Server SDK (Unreal) 5.x: azioni

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift Unreal per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#) e per informazioni sull'utilizzo del plug-in del server Unreal SDK, consulta. [Integra Amazon GameLift in un progetto Unreal Engine](#)

Azioni

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetTerminationTime\(\)](#)

- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)

Note

Questo argomento descrive l'API Amazon GameLift 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`

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 SDK per una flotta EC2 gestita. Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo legge i

parametri del server dall'ambiente host per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
FGameLiftGenericOutcome InitSDK()
```

Valore restituito

In caso di successo, restituisce un `InitSdkOutcome` oggetto che indica che il processo del server è pronto per la chiamata [ProcessReady\(\)](#).

Esempio

```
//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK();
```

InitSDK()

Inizializza l'Amazon GameLift SDK per una flotta. Anywhere Chiama questo metodo all'avvio, prima che GameLift si verifichi qualsiasi altra inizializzazione relativa ad Amazon. Questo metodo richiede parametri espliciti del server per configurare la comunicazione tra il server e il GameLift servizio Amazon.

Sintassi

```
FGameLiftGenericOutcome InitSDK(serverParameters)
```

Parametri

[F ServerParameters](#)

Per inizializzare un server di gioco su una GameLift Anywhere flotta Amazon, costruisci un `ServerParameters` oggetto con le seguenti informazioni:

- L'URL WebSocket usato per connetterti al tuo 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 GameLift flotta Amazon contenente i tuoi dati di GameLift Anywhere calcolo Amazon.

- Il token di autorizzazione generato dall' GameLift operazione Amazon.

Valore restituito

In caso di successo, restituisce un `InitSdkOut` come oggetto che indica che il processo del server è pronto per la chiamata [ProcessReady\(\)](#).

Note

Se le chiamate a `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 GameLift agent to enable
further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK(serverParameters);
```

ProcessReady()

Notifica ad Amazon GameLift 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 [F ProcessParameters](#) oggetto che comunica le seguenti informazioni sul processo del server:

- Nomi dei metodi di callback implementati nel codice del server di gioco richiamato dal GameLift servizio Amazon 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 che Amazon GameLift acquisisca e archivi.

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 GameLift 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 ad Amazon GameLift che il processo del server sta terminando. Richiama 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. GameLift 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 GameLift we are indeed going to shutdown.
params->OnTerminate.BindLambda( [=]() {
    UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
    gameLiftSdkModule->ProcessEnding();
});
```

ActivateGameSession()

Notifica ad Amazon GameLift che il processo del server ha attivato una sessione di gioco ed è ora pronto per ricevere le connessioni dei giocatori. Questa azione deve essere richiamata come parte della funzione di `onStartGameSession()` 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 `onStartGameSession()` delegata.

```
//When a game session is created, GameLift 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 gameId = FString(gameSession.GetGameSessionId());
    UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameId);
```

```
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 "F GameLiftError"](#)

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, GameLift 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 gameId = FString(gameSession.GetGameSessionId());
    UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameId);
    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. Amazon GameLift chiama `onProcessTerminate()` per i seguenti motivi:

- Quando il processo del server ha segnalato problemi di salute o non ha risposto ad Amazon GameLift.
- Quando si interrompe 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 [Rispondere a una notifica di arresto del processo del server](#)

Esempio

```
AwsDateTimeOutcome TermTimeOutcome = gameLiftSdkModule->GetTerminationTime();
```

AcceptPlayerSession()

Notifica ad Amazon GameLift che un giocatore con l'ID di sessione del giocatore specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID di sessione del giocatore sia valido. Dopo la convalida della sessione di gioco, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO ad ATTIVO.

Sintassi

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerId)
```

Parametri

playerSessionId

ID univoco rilasciato da Amazon 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 gestisce una richiesta di connessione che include la convalida e il rifiuto di ID di sessione giocatore non validi.

```
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 ad Amazon GameLift che un giocatore si è disconnesso dal processo del server. In risposta, Amazon GameLift cambia lo slot del giocatore rendendolo disponibile.

Sintassi

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerSessionId)
```


Parametri

playerSessionId

ID univoco rilasciato da Amazon 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

```
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

[F GameLiftDescribePlayerSessionsRequest](#)

Un [the section called “F GameLiftDescribePlayerSessionsRequest”](#) oggetto che descrive le sessioni dei giocatori da recuperare.

Valore restituito

Se l'esito è positivo, restituisce un oggetto [the section called “F GameLiftDescribePlayerSessionsOutcome”](#) 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 NextToken e impostando il valore Limite su 10, Amazon GameLift restituisce i primi 10 record delle sessioni di gioco 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 [FlexMatch riempimento](#).

Questa operazione è asincrona. Se vengono abbinati nuovi giocatori, Amazon GameLift fornisce dati aggiornati sui 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

[F 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, Amazon ne GameLift 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

[F 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

```
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 tua GameLift Anywhere risorsa di elaborazione Amazon e Amazon. GameLift Puoi utilizzare il percorso del certificato quando registri il tuo dispositivo di elaborazione in una GameLift Anywhere flotta Amazon. Per ulteriori informazioni, consulta, [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 GameLift Amazon 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

[F GameLiftGetFleetRoleCredentialsRequest](#)

Valore restituito

Restituisce un oggetto [the section called “F GameLiftGetFleetRoleCredentialsOutcome”](#).

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
}

```

Riferimento all'SDK per GameLift server Amazon (Unreal): tipi di dati

Puoi utilizzare questo riferimento all'SDK del server Amazon GameLift Unreal per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#) e per informazioni sull'utilizzo del plug-in del server Unreal SDK, consulta. [Integra Amazon GameLift in un progetto Unreal Engine](#)

Tipi di dati

- [F ProcessParameters](#)

- [UpdateGameSession](#)
- [GameSession](#)
- [F ServerParameters](#)
- [F StartMatchBackfillRequest](#)
- [FPlayer](#)
- [F GameLiftDescribePlayerSessionsRequest](#)
- [F StopMatchBackfillRequest](#)
- [F AttributeValue](#)
- [F GameLiftGetFleetRoleCredentialsRequest](#)
- [F GameLiftLongOutcome](#)
- [F GameLiftStringOutcome](#)
- [F GameLiftDescribePlayerSessionsOutcome](#)
- [F GameLiftDescribePlayerSessionsResult](#)
- [F GenericOutcome](#)
- [F GameLiftPlayerSession](#)
- [F GameLiftGetComputeCertificateOutcome](#)
- [F GameLiftGetComputeCertificateResult](#)
- [F GameLiftGetFleetRoleCredentialsOutcome](#)
- [F GetFleetRoleCredentialsResult](#)
- [F GameLiftError](#)
- [enumerazioni;](#)

Note

Questo argomento descrive l'API Amazon GameLift 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`

F ProcessParameters

Questo tipo di dati contiene il set di parametri inviati ad Amazon GameLift in un [ProcessReady\(\)](#).

Proprietà	Descrizione
LogParameters	<p>Un oggetto con percorsi di directory ai file generati durante una sessione di gioco. Amazon GameLift copia e archivia i file per accessi futuri.</p> <p>Tipo: TArray<FString></p> <p>Required: No</p>
OnHealthCheck	<p>La funzione di callback GameLift richiamata da Amazon per richiedere un rapporto sullo stato di salute del processo del server. Amazon GameLift chiama 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 registra il processo del server come non integro.</p> <p>Questa proprietà è una funzione delegata definita come <code>DECLARE_DELEGATE_RetVal(bool, FOnHealthCheck)</code> ;</p> <p>Tipo: FOnHealthCheck</p> <p>Required: No</p>
OnProcessTerminate	<p>La funzione di callback GameLift richiamata da Amazon per forzare l'arresto del processo server. Dopo aver chiamato questa funzione, Amazon GameLift attende 5 minuti che il processo server si spenga e risponda con una ProcessEnding() chiamata prima di chiudere il processo server.</p> <p>Tipo: FSimpleDelegate</p>

<p>OnStartGameSession</p>	<p>Campo obbligatorio: sì</p> <p>La funzione di callback che Amazon GameLift richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta a una richiesta del cliente CreateGameSession. La funzione di callback passa un GameSession oggetto, come definito nell'Amazon GameLift API Reference.</p> <p>Questa proprietà è una funzione delegata definita come <code>DECLARE_DELEGATE_OneParam(FOnStartGameSession, Aws::GameLift::Server::Model::GameSession);</code></p> <p>Tipo: <code>FOnStartGameSession</code></p> <p>Campo obbligatorio: sì</p>
<p>OnUpdateGameSession</p>	<p>La funzione di callback GameLift richiamata da Amazon per passare un oggetto di sessione di gioco aggiornato al processo del server. Amazon GameLift chiama questa funzione quando viene elaborata una richiesta di match backfill per fornire dati aggiornati sul matchmaker. Passa un GameSession oggetto, uno status update (<code>updateReason</code>) e l'ID del ticket match backfill.</p> <p>Questa proprietà è una funzione delegata definita come <code>DECLARE_DELEGATE_OneParam(FOnUpdateGameSession, Aws::GameLift::Server::Model::UpdateGameSession);</code></p> <p>Tipo: <code>FOnUpdateGameSession</code></p> <p>Required: No</p>

<p>Porta</p>	<p>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.</p> <p>Tipo: <code>int</code></p> <p>Campo obbligatorio: sì</p>
--------------	--

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
<p>GameSession</p>	<p>Un GameSession oggetto definito dall' GameLift API Amazon. L'GameSession oggetto contiene proprietà che descrivono una sessione di gioco.</p> <p>Tipo: <code>Aws::GameLift::Server::GameSession</code></p> <p>Required: No</p>
<p>UpdateReason</p>	<p>Il motivo per cui la sessione di gioco viene aggiornata.</p> <p>Tipo: <code>enum class UpdateReason</code></p> <ul style="list-style-type: none"> • <code>MATCHMAKING_DATA_UPDATED</code>

Proprietà	Descrizione
	<ul style="list-style-type: none"> • BACKFILL_FAILED • BACKFILL_TIMED_OUT • BACKFILL_ANNULLATO <p>Required: No</p>
BackfillTicketId	<p>L'ID del ticket di backfill che tenta di aggiornare la sessione di gioco.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>

GameSession

Questo tipo di dati fornisce i dettagli di una sessione di gioco.

Proprietà	Descrizione
GameSessionId	<p>Un identificatore univoco per la sessione di gioco. L'ARN di una sessione di gioco ha il seguente formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code></p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>
Nome	<p>Un'etichetta descrittiva della sessione di gioco.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>

Proprietà	Descrizione
FleetId	<p>Un identificatore univoco per la flotta su cui è in esecuzione la sessione di gioco.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>
MaximumPlayerSessionCount	<p>Il numero massimo di connessioni dei giocatori alla sessione di gioco.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>
Porta	<p>Il numero di porta per la sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>
IpAddress	<p>L'indirizzo IP della sessione di gioco. Per connettersi a un server di GameLift gioco Amazon, un'app necessita sia dell'indirizzo IP che del numero di porta.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>
GameSessionData	<p>Un set di proprietà personalizzate della sessione di gioco, formattate come valore di stringa singola.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>

Proprietà	Descrizione
MatchmakerData	<p>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.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>
GameProperties	<p>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.</p> <p>Tipo: <code>GameProperty[]</code></p> <p>Required: No</p>

Proprietà	Descrizione
DnsName	<p>L'identificatore DNS assegnato all'istanza che esegue la sessione di gioco. I valori hanno il seguente formato:</p> <ul style="list-style-type: none"> • Flotte abilitate per TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> • Flotte non compatibili con TLS: <code>ec2-<unique identifier>.compute.amazonaws.com</code> <p>Quando ti connetti a una sessione di gioco in esecuzione su una flotta abilitata per TLS, devi utilizzare il nome DNS, non l'indirizzo IP.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>

F ServerParameters

Informazioni utilizzate per mantenere la connessione tra un GameLift Anywhere server Amazon e il GameLift servizio Amazon. Queste informazioni vengono utilizzate quando si avviano nuovi processi server con [InitSDK\(\)](#). Per i server ospitati su istanze EC2 GameLift gestite da Amazon, usa un oggetto vuoto.

Proprietà	Descrizione
websocketUrl	<p><code>GameLiftServerSdkEndpoint</code> Amazon GameLift restituisce quando RegisterCompute richiedi una risorsa di GameLift Anywhere elaborazione Amazon.</p> <p>Tipo: <code>char[]</code></p>

Proprietà	Descrizione
	Campo obbligatorio: sì
ID del processo	<p>Un identificatore univoco registrato nel processo del server che ospita il gioco.</p> <p>Tipo: <code>char []</code></p> <p>Campo obbligatorio: sì</p>
hostId	<p>HostIDViene ComputeName utilizzato al momento della registrazione del computer. Per ulteriori informazioni, vedere, RegisterCompute.</p> <p>Tipo: <code>char []</code></p> <p>Campo obbligatorio: sì</p>
FleetID	<p>L'identificatore univoco della flotta su cui è registrato il computer. Per ulteriori informazioni, vedere,. RegisterCompute</p> <p>Tipo: <code>char []</code></p> <p>Campo obbligatorio: sì</p>
AuthToken	<p>Il token di autenticazione generato da Amazon GameLift che autentica il tuo server su Amazon. GameLift Per ulteriori informazioni, consulta,. GetComputeAuthToken</p> <p>Tipo: <code>char []</code></p> <p>Campo obbligatorio: sì</p>

F StartMatchBackfillRequest

Informazioni utilizzate per creare una richiesta di riempimento del matchmaking. Il server di gioco comunica queste informazioni ad Amazon GameLift tramite una [StartMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>Un identificatore di sessione di gioco univoco. L'operazione API GetGameSessionId restituisce l'identificatore in formato ARN.</p> <p>Tipo: <code>char[]</code></p> <p>Campo obbligatorio: sì</p>
MatchmakingConfigurationArn	<p>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 <code>matchmaker data</code>. Scopri di più sui dati dei matchmaker in Work with matchmaker data.</p> <p>Tipo: <code>char[]</code></p> <p>Campo obbligatorio: sì</p>
Players	<p>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 rappresentino un buon abbinamento per i giocatori attuali.</p> <p>Tipo: <code>TArray<FPlayer></code></p> <p>Campo obbligatorio: sì</p>
TicketId	<p>Un identificatore univoco per un ticket di richiesta di matchmaking o match backfill. Se non fornisci un valore, Amazon ne GameLift genera uno. Utilizzare questo identificatore per monitorare lo stato del ticket di backfill</p>

Proprietà	Descrizione
	<p>degli abbinamenti o annullare la richiesta, se necessario.</p> <p>Tipo: <code>char[]</code></p> <p>Required: No</p>

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 aggiunge le informazioni sulla squadra dopo la fine di una partita.

Proprietà	Descrizione
LatencyInSM	<p>Un insieme di valori espressi in millisecondi che indica la quantità di latenza che un giocatore sperimenta quando è connesso a una posizione.</p> <p>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.</p> <p>Tipo: <code>TMap>FString, int32<</code></p> <p>Required: No</p>
PlayerAttributes	<p>Una raccolta di coppie chiave:valore contenenti informazioni sui giocatori da utilizzare nel matchmaking. Le chiavi degli attributi del giocatore devono corrispondere a quelle <code>PlayerAttributes</code> utilizzate in un set di regole di matchmaking.</p>

Proprietà	Descrizione
	<p>Per ulteriori informazioni sugli attributi dei giocatori, vedi AttributeValue.</p> <p>Tipo: TMap>FString, FAttributeValue<</p> <p>Required: No</p>
PlayerId	<p>Un identificatore univoco per un giocatore.</p> <p>Tipo: std::string</p> <p>Required: No</p>
Team	<p>Il nome della squadra a cui è assegnato il giocatore in una partita. Definisci il nome della squadra nel set di regole del matchmaking.</p> <p>Tipo: FString</p> <p>Required: No</p>

F GameLiftDescribePlayerSessionsRequest

Un oggetto che specifica le sessioni dei giocatori da recuperare. Il processo del server fornisce queste informazioni con una [DescribePlayerSessions\(\)](#) chiamata ad Amazon GameLift.

Proprietà	Descrizione
GameSessionId	<p>Un identificatore di sessione di gioco univoco. Utilizzare questo parametro per richiedere tutte le sessioni giocatore per la sessione di gioco specificata.</p> <p>Il formato dell'ID della sessione di gioco èFString. GameSessionID È una stringa ID personalizzata o un</p>

Proprietà	Descrizione
	Tipo: <code>std::string</code> Required: No
PlayerSessionId	L'identificatore univoco per la sessione di un giocatore. Usa questo parametro per richiedere una singola sessione di gioco specifica. Tipo: <code>FString</code> Required: No
PlayerId	L'identificatore univoco di un giocatore. Usa questo parametro per richiedere tutte le sessioni di gioco per un giocatore specifico. Per informazioni, consulta Genera gli ID dei giocatori . Tipo: <code>FString</code> Required: No

Proprietà	Descrizione
PlayerSessionStatusFilter	<p>Lo stato della sessione del giocatore su cui filtrare i risultati. I possibili stati delle sessioni di gioco includono:</p> <ul style="list-style-type: none">• 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). <p>Tipo: FString</p> <p>Required: No</p>
NextToken	<p>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.</p> <p>Tipo: FString</p> <p>Required: No</p>

Proprietà	Descrizione
Limite	<p>Numero massimo di risultati da restituire. Se fornisci un ID di sessione del giocatore, questo parametro viene ignorato.</p> <p>Tipo: <code>int</code></p> <p>Required: No</p>

F StopMatchBackfillRequest

Informazioni utilizzate per annullare una richiesta di ripristino del matchmaking. Il server di gioco comunica queste informazioni al GameLift servizio Amazon tramite una [StopMatchBackfill\(\)](#) chiamata.

Proprietà	Descrizione
GameSessionArn	<p>Un identificatore univoco della sessione di gioco della richiesta annullata.</p> <p>Tipo: <code>FString</code></p> <p>Campo obbligatorio: sì</p>
MatchmakingConfigurationArn	<p>Un identificatore univoco del matchmaker a cui è stata inviata questa richiesta.</p> <p>Tipo: <code>FString</code></p> <p>Campo obbligatorio: sì</p>
TicketId	<p>Un identificatore univoco del ticket di richiesta di riempimento da annullare.</p> <p>Tipo: <code>FString</code></p> <p>Campo obbligatorio: sì</p>

F AttributeValue

Usa questi valori nelle coppie chiave-valore degli attributi. [FPlayer](#) 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	<p>Specifica il tipo di valore dell'attributo.</p> <p>Tipo: un valore <code>FAttributeType</code> enum.</p> <p>Required: No</p>
S	<p>Rappresenta un valore di attributo di tipo stringa.</p> <p>Tipo: <code>FString</code></p> <p>Required: No</p>
N	<p>Rappresenta un valore di attributo numerico.</p> <p>Tipo: <code>double</code></p> <p>Required: No</p>
SL	<p>Rappresenta una matrice di valori di attributi di tipo stringa.</p> <p>Tipo: <code>TArray<FString></code></p> <p>Required: No</p>
SDM	<p>Rappresenta un dizionario di chiavi di stringa e valori doppi.</p> <p>Tipo: <code>TMap<FString, double></code></p> <p>Required: No</p>

F GameLiftGetFleetRoleCredentialsRequest

Questo tipo di dati fornisce credenziali di ruolo che estendono l'accesso limitato alle tue AWS risorse al server di gioco. Per ulteriori informazioni, consulta [Configura un ruolo di servizio IAM per Amazon GameLift](#).

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

F GameLiftLongOutcome

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 "F GameLiftError" Required: No

F GameLiftStringOutcome

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

Proprietà	Descrizione
	Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo. Tipo: the section called “F GameLiftError” Required: No

F GameLiftDescribePlayerSessionsOutcome

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 “F GameLiftDescribePlayerSessionsResult” Required: No
ResultWithOwnership	Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto. Tipo: <code>FGameLiftDescribePlayerSessionsResult&&</code> Required: No
Riuscito	Se l'azione ha avuto successo o meno. Tipo: <code>bool</code> Campo obbligatorio: sì
Errore	L'errore che si è verificato se l'azione non ha avuto successo.

Proprietà	Descrizione
	Tipo: the section called “F GameLiftError” Required: No

F GameLiftDescribePlayerSessionsResult

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 “F GameLiftError” Required: No

F GenericOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "F GameLiftError"</p> <p>Required: No</p>

F GameLiftPlayerSession

Proprietà	Descrizione
CreationTime	<p>Tipo: long</p> <p>Campo obbligatorio: sì</p>
FleetId	<p>Tipo: FString</p> <p>Campo obbligatorio: sì</p>
GameSessionId	<p>Tipo: FString</p> <p>Campo obbligatorio: sì</p>
IpAddress	<p>Tipo: FString</p> <p>Campo obbligatorio: sì</p>
PlayerData	<p>Tipo: FString</p>

Proprietà	Descrizione
	Campo obbligatorio: sì
PlayerId	Tipo: FString Campo obbligatorio: sì
PlayerSessionId	Tipo: FString Campo obbligatorio: sì
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

F GameLiftGetComputeCertificateOutcome

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 “F GameLiftGetComputeCertificateResult” Required: No

Proprietà	Descrizione
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: <code>FGameLiftGetComputeCertificateResult&&</code></p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "F GameLiftError"</p> <p>Required: No</p>

F GameLiftGetComputeCertificateResult

Il percorso del certificato TLS sul computer e il nome host del computer.

Proprietà	Descrizione
CertificatePath	<p>Tipo: <code>FString</code></p> <p>Campo obbligatorio: sì</p>
ComputeName	<p>Tipo: <code>FString</code></p> <p>Campo obbligatorio: sì</p>

F GameLiftGetFleetRoleCredentialsOutcome

Questo tipo di dati risulta da un'azione e produce un oggetto con le seguenti proprietà:

Proprietà	Descrizione
Risultato	<p>Il risultato dell'azione.</p> <p>Tipo: the section called “F GetFleetRoleCredentialsResult”</p> <p>Required: No</p>
ResultWithOwnership	<p>Il risultato dell'azione, espresso come rvalue, in modo che il codice chiamante possa assumere la proprietà dell'oggetto.</p> <p>Tipo: <code>FGameLiftGetFleetRoleCredentialsResult&&</code></p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: <code>bool</code></p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called “F GameLiftError”</p> <p>Required: No</p>

F GetFleetRoleCredentialsResult

Proprietà	Descrizione
AccessKeyId	L'ID della chiave di accesso per autenticare e fornire l'accesso alle tue risorse. AWS 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 Required: No
Expiration	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

Proprietà	Descrizione
SessionToken	<p>Un token per identificare la sessione attiva corrente che interagisce con AWS le tue risorse.</p> <p>Tipo: FString</p> <p>Required: No</p>
Riuscito	<p>Se l'azione ha avuto successo o meno.</p> <p>Tipo: bool</p> <p>Campo obbligatorio: sì</p>
Errore	<p>L'errore che si è verificato se l'azione non ha avuto successo.</p> <p>Tipo: the section called "GameLiftError"</p> <p>Required: No</p>

F GameLiftError

Proprietà	Descrizione
ErrorType	<p>Il tipo di errore.</p> <p><u>Tipo: Un enum. GameLiftErrorType</u></p> <p>Required: No</p>
ErrorMessage	<p>Il nome dell'errore.</p> <p>Tipo: std::string</p> <p>Required: No</p>
ErrorMessage	<p>Messaggio di errore.</p>

Proprietà	Descrizione
	Tipo: <code>std::string</code> Required: No

enumerazioni;

Le enumerazioni definite per l'Amazon GameLift server SDK (Unreal) sono definite come segue:

F AttributeType

- 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 ad Amazon non è riuscita. GameLift
- 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 server o il client GameLift Amazon è già stato inizializzato con `Initialize ()`.
- FLEET_MISMATCH — La flotta di destinazione non corrisponde alla flotta di una `GameSession` o `PlayerSession`.
- GAMELIFT_CLIENT_NOT_INITIALIZED — Il client Amazon non è stato inizializzato. GameLift
- GAMELIFT_SERVER_NOT_INITIALIZED — Il server Amazon non è stato inizializzato. GameLift
- GAME_SESSION_ENDED_FAILED — L'SDK di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare la fine della sessione di gioco.

- `GAME_SESSION_NOT_READY` — La sessione di gioco di GameLift Amazon Server non è stata attivata.
- `GAME_SESSION_READY_FAILED` — L'SDK di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare che la sessione di gioco è pronta.
- `INITIALIZATION_MISMATCH` — È stato chiamato un metodo client dopo `Server: :Initialize ()` o viceversa.
- `NOT_INITIALIZED` — Il GameLift server o il client Amazon non è stato inizializzato 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 di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare la fine del processo.
- `PROCESS_NOT_ACTIVE` — Il processo del server non è ancora attivo, non è legato 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 di GameLift Amazon Server non è riuscito a contattare il servizio per segnalare che il processo è pronto.
- `SDK_VERSION_DETECTION_FAILED` — Rilevamento della versione SDK non riuscito.
- `STX_CALL_FAILED` — Una chiamata al componente backend del server `xSTx` non è riuscita.
- `STX_INITIALIZATION_FAILED` — L'inizializzazione del componente di backend del server `XStx` non è riuscita.
- `UNEXPECTED_PLAYER_SESSION` — Il server ha rilevato una sessione 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.

E `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.

E `PlayerSessionStatus`

- `ACTIVE`
- `COMPLETED`
- `NON IMPOSTATO`
- `RISERVATO`
- `TIMEOUT`

Riferimento all'SDK GameLift 3.x del server Amazon Unreal Engine

Puoi usare questo riferimento all'SDK 3.x del server Amazon GameLift Unreal Engine per aiutarti a preparare il tuo gioco multiplayer da utilizzare con Amazon. GameLift Per informazioni dettagliate sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Argomenti

- [Riferimento SDK GameLift del server Amazon per Unreal Engine: azioni](#)
- [Riferimento Amazon GameLift Server SDK per Unreal Engine: tipi di dati](#)

Riferimento SDK GameLift del server Amazon per Unreal Engine: azioni

Questo riferimento all'SDK di Amazon GameLift Server può aiutarti a preparare i tuoi progetti di gioco Unreal Engine da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Questa API è definita in `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Per configurare il plugin Unreal Engine e vedere esempi di codice [Integra Amazon GameLift in un progetto Unreal Engine](#).

- Operazioni
- [Tipi di dati](#)

AcceptPlayerSession()

Notifica al GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è connesso al processo del server e deve essere convalidato. Amazon GameLift verifica che l'ID della sessione del giocatore sia valido, ovvero che l'ID giocatore abbia riservato uno slot per il giocatore nella sessione di gioco. Una volta convalidato, Amazon GameLift modifica lo stato dello slot del giocatore da RISERVATO a ATTIVO.

Sintassi

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerId)
```

Parametri

playerSessionId

ID univoco emesso dal GameLift servizio Amazon in risposta a una chiamata all'azione AWS [CreatePlayerSession](#) SDK Amazon GameLift API. 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 al GameLift servizio Amazon che il processo del server ha attivato una sessione di gioco ed è ora 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

```
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 [F DescribePlayerSessionsRequest](#) che descrive le sessioni giocatore da recuperare.

Campo obbligatorio: sì

Valore restituito

Se l'esito è positivo, restituisce un oggetto [F DescribePlayerSessionsRequest](#) 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 [PlayerSession](#) dati dell'GameLiftAPI Amazon AWS SDK.

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 al parco veicoli e alle relative istanze. AWS Certificate Manager genera questo certificato quando si crea una nuova flotta 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 certificati TLS e della catena di certificati della flotta, che sono archiviati nell'istanza. Nell'istanza viene inoltre archiviato un file di certificato principale, 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 [GetCertificateResponse Elements](#) in the 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 l'Amazon GameLift SDK. Questo metodo deve essere richiamato all'avvio, prima che si verifichi qualsiasi altra inizializzazione GameLift relativa ad Amazon.

Sintassi

```
FGameLiftGenericOutcome InitSDK()
```

Parametri

Questa operazione non prevede parametri.

Valore restituito

Restituisce un risultato generico composto da successo o fallimento con un messaggio di errore.

ProcessEnding()

Notifica al GameLift servizio Amazon che il processo del server si sta spegnendo. 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 composto da successo o fallimento con un messaggio di errore.

ProcessReady()

Notifica al GameLift servizio Amazon 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

F ProcessParameters

Un oggetto [F ProcessParameters](#) che comunica le informazioni seguenti sul processo del server:

- Nomi dei metodi di callback, implementati nel codice del server di gioco, che il GameLift servizio Amazon 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 che Amazon GameLift acquisisca e memorizzi.

Campo obbligatorio: sì

Valore restituito

Restituisce un risultato generico composto 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 al GameLift servizio Amazon che un giocatore con l'ID di sessione specificato si è disconnesso dal processo del server. In risposta, Amazon GameLift modifica lo slot del giocatore rendendolo disponibile, il che consente di assegnarlo a un nuovo giocatore.

Sintassi

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerId)
```

Parametri

playerSessionId

ID univoco emesso dal GameLift servizio Amazon in risposta a una chiamata all'azione AWS [CreatePlayerSession](#) SDK Amazon GameLift API. 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. Scopri di più sulla [funzione di FlexMatch riempimento](#).

Questa operazione è asincrona. Se i nuovi giocatori vengono abbinati con successo, il GameLift servizio Amazon 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

F StartMatchBackfillRequest

Un oggetto [F StartMatchBackfillRequest](#) 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 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 SDK [StopMatchmaking\(\)](#). Scopri di più sulla [funzione di FlexMatch riempimento](#).

Sintassi

```
FGameLiftGenericOutcome StopMatchBackfill (FStopMatchBackfillRequest  
&stopBackfillRequest);
```

Parametri

StopMatchBackfillRequest

Un oggetto [F 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.

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 GameLift servizio Amazon che il processo del server ha terminato la sessione di gioco corrente. Questa azione viene richiamata quando il processo del server rimarrà attivo e pronto per ospitare una nuova sessione di gioco. Dovrebbe essere richiamato solo dopo il completamento della procedura di chiusura della sessione di gioco, poiché segnala ad Amazon GameLift che il processo del server è immediatamente disponibile per ospitare una nuova sessione di gioco.

Questa azione non viene chiamata se il processo del server verrà chiuso dopo l'interruzione della sessione di gioco. Chiama invece [ProcessEnding\(\)](#) per segnalare che sia la sessione di gioco che il processo del 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 nell'Amazon GameLift Service API Reference).

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.

Riferimento Amazon GameLift Server SDK per Unreal Engine: tipi di dati

Questa guida di riferimento all'SDK di Amazon GameLift Server può aiutarti a preparare i tuoi progetti di gioco Unreal Engine da utilizzare con Amazon. GameLift Per i dettagli sul processo di integrazione, consulta [Aggiungi Amazon GameLift al tuo server di gioco](#).

Questa API è definita in `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Per configurare il plugin Unreal Engine e vedere esempi di codice [Integra Amazon GameLift in un progetto Unreal Engine](#).

- [Operazioni](#)
- Tipi di dati

F DescribePlayerSessionsRequest

Questo tipo di dati viene utilizzato per specificare quale sessione del giocatore recuperare. Puoi utilizzarlo come segue:

- Fornisci un `PlayerSessionId` modulo per richiedere una sessione specifica per il giocatore.
- Fornisci un `GameSessionId` per richiedere tutte le sessioni dei giocatori nella sessione di gioco specificata.
- Fornisci un `PlayerId` 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 `NextToken` per ottenere risultati sotto forma di un insieme 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. Gli ID giocatore sono definiti dallo sviluppatore. Consultare [Genera gli ID dei giocatori](#).

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

F ProcessParameters

Questo tipo di dati contiene il set di parametri inviati al GameLift servizio Amazon durante 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 il GameLift servizio Amazon richiama per attivare una nuova sessione di gioco. Amazon GameLift chiama questa funzione in risposta alla richiesta del cliente [CreateGameSession](#). La funzione di callback accetta un [GameSession](#) oggetto (definito nell'Amazon GameLift Service API Reference).

Tipo: F OnStartGameSession

Campo obbligatorio: sì

onProcessTerminate

Nome della funzione di callback che il GameLift servizio Amazon richiama per forzare l'arresto del processo del server. Dopo aver richiamato questa funzione, Amazon GameLift attende cinque minuti che il processo del server si spenga e risponda con una [ProcessEnding\(\)](#) chiamata prima di chiudere il processo del server.

Tipo: F SimpleDelegate

Campo obbligatorio: no

onHealthCheck

Nome della funzione di callback che il GameLift servizio Amazon richiama per richiedere un rapporto sullo stato di salute dal processo del server. Amazon GameLift chiama questa funzione ogni 60 secondi. Dopo aver richiamato questa funzione, Amazon GameLift attende 60 secondi per ricevere una risposta e, se non ne riceve alcuna, registra il processo del server come non integro.

Tipo: F OnHealthCheck

Campo obbligatorio: no

onUpdateGameSessione

Nome della funzione di callback che il GameLift servizio Amazon richiama per passare un oggetto della sessione di gioco aggiornato al processo del server. Amazon GameLift richiama questa funzione quando viene elaborata una richiesta di [backfill della partita](#) per fornire dati aggiornati sul matchmaker. Passa un [GameSession](#) oggetto, un aggiornamento di stato (updateReason) e l'ID del ticket di backfill corrispondente.

Tipo: F OnUpdateGameSession

Campo obbligatorio: no

F StartMatchBackfillRequest

Questo tipo di dati viene utilizzato per inviare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al GameLift servizio Amazon durante 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 del matchmaker in [Lavora con i dati del 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. Consulta la Amazon GameLift API Reference Guide per una descrizione del formato degli oggetti di Player. Per trovare gli attributi dei giocatori, gli ID e gli incarichi delle squadre, osservare l'oggetto delle sessioni di gioco nella proprietà dei dati del matchmaker. Se la latenza viene utilizzata dal matchmaker, raccogliere la latenza aggiornata per la regione attuale e includerla nei dati di ciascun giocatore.

Tipo: 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 ne GameLift 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

F StopMatchBackfillRequest

Questo tipo di dati viene utilizzato per annullare una richiesta di backfill di abbinamento. Le informazioni vengono comunicate al GameLift servizio Amazon durante 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

Eventi di collocamento delle sessioni di gioco

Amazon GameLift 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 posizionamento delle sessioni di gioco, [GameSessionPlacement](#) consulta Amazon GameLift API Reference.

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 utilizzato per gestire Amazon GameLift.

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`

Flotte non abilitate per TLS: `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 `inGameSessionArn`.

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-1232131232324124123123"
    }
  ]
}
}

```

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 non GameLift 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"
  }
}
```

Generazione di stime GameLift dei prezzi di Amazon

Con AWS Pricing Calculator, puoi [creare una stima dei prezzi per Amazon GameLift](#). Non è necessaria una conoscenza Account AWS o una conoscenza approfondita AWS per utilizzare la calcolatrice.

AWS Pricing Calculator Il calcolatore ti guida attraverso le decisioni che influiscono sui costi del servizio per darti un'idea di quanto Amazon GameLift potrebbe costare per il tuo progetto di gioco. Se non sei ancora sicuro di come intendi utilizzare Amazon GameLift, utilizza i valori predefiniti per generare una stima. Durante la pianificazione dell'utilizzo della produzione, la calcolatrice può aiutarti a testare potenziali scenari e generare stime più accurate.

Puoi utilizzarlo AWS Pricing Calculator per generare stime per le seguenti opzioni GameLift di hosting Amazon:

- [Stima dell'GameLift hosting Amazon](#)
- [Stima Amazon GameLift standalone FlexMatch](#)

Stima dell'GameLift hosting Amazon

Questa opzione fornisce una stima dei costi per l'hosting dei giochi su server GameLift gestiti di Amazon, inclusi i costi per l'utilizzo delle istanze del server e il trasferimento dei dati. FlexMatch il matchmaking è incluso nel costo dell'hosting GameLift gestito da Amazon.

Se stai ospitando 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.

GameLift Istanze Amazon

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 utilizza [istanze Amazon Elastic Compute Cloud \(Amazon EC2\) per gestire i server](#) di gioco. In Amazon GameLift, distribuisce una flotta di istanze con un tipo di istanza e un sistema operativo specifici. Se hai o prevedi di avere più flotte, crea una stima per ciascuna flotta.

Per iniziare, apri la [GameLift pagina Configura Amazon](#) di AWS Pricing Calculator. Aggiungi una descrizione, scegli una regione, quindi scegli Estimate l'GameLift hosting Amazon (Instance + Data Transfer Out). In GameLift Istanze Amazon, completa i seguenti campi:

- Numero massimo di giocatori simultanei (CCU di picco)

Questo è il numero massimo di giocatori che possono connettersi ai tuoi server di gioco contemporaneamente. Questo campo indica la capacità di hosting necessaria ad Amazon per GameLift soddisfare il picco della domanda dei giocatori. Inserisci il numero massimo giornaliero di giocatori che prevedi di ospitare utilizzando 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 deve mantenere per i tuoi giocatori. Se non sei sicuro del valore percentuale da utilizzare, mantieni il valore predefinito di **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 tuo gioco ha una CCU oraria media di 6.000 e una CCU di picco di 10.000, inserisci il valore della percentuale. **60**

- Sessioni di gioco per istanza

Questo è il numero di sessioni di gioco che ciascuna istanza del tuo server di gioco può ospitare contemporaneamente. I fattori che possono influire su questo numero includono i requisiti di risorse del server di gioco, il numero di giocatori da ospitare in ogni sessione di gioco e le aspettative 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 hai modalità di gioco con un numero diverso di giocatori, stima il numero medio di giocatori per sessione di gioco nell'intero gioco. Il valore di default è **8**.

- Buffer inattivo dell'istanza%

Questa è la percentuale di capacità di hosting non utilizzata da mantenere in riserva per gestire i picchi improvvisi della domanda 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 inattivo del 20%, una flotta che supporta giocatori con 100 istanze attive mantiene 20 istanze inattive.

- Istanza Spot%

GameLiftLe flotte Amazon 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 conveniente. Si consiglia di utilizzare una combinazione per ottimizzare sia i risparmi sui costi che la disponibilità. Per informazioni su come Amazon GameLift utilizza le istanze Spot, consulta [Istanze On-Demand e istanze Spot](#).

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

Ad esempio, se distribuisce una flotta di 100 istanze e specifichi una **40** percentuale, Amazon si GameLift impegna a mantenere 60 istanze on demand e 40 istanze spot.

- Tipo di istanza

GameLiftLe flotte Amazon possono utilizzare una gamma di tipi di istanze Amazon EC2 che variano in termini di potenza di calcolo, memoria, storage e funzionalità di rete. Quando configuri una GameLift flotta Amazon, scegli il tipo di istanza più adatto alle esigenze del tuo gioco. Per informazioni sulla selezione di un tipo di istanza con AmazonGameLift, consulta [Scelta delle GameLift risorse di elaborazione Amazon](#).

Se conosci il tipo di istanza che stai utilizzando o intendi utilizzare nel tuo GameLift parco istanze Amazon, scegli quel tipo. Se non sei sicuro del tipo da scegliere, considera 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 vengono eseguiti 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 [GameLift pagina Configura Amazon](#) di AWS Pricing Calculator, 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 per il tuo gioco.

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

Nel campo seguente, stimate 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 ciascuna 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 in KB/sec. Il valore di default è **4**.

Quando hai finito di configurare la stima GameLift dei prezzi di Amazon, scegli **Aggiungi alla mia stima**. Per ulteriori informazioni sulla creazione e la gestione delle stime in AWS Pricing Calculator, consulta [Creare un preventivo, configurare un servizio e aggiungere altri servizi](#) nella Guida per l'AWS Pricing Calculator.

Stima Amazon GameLift standalone FlexMatch

Questa opzione fornisce una stima dei costi per l'utilizzo del FlexMatch matchmaking come servizio autonomo mentre si ospitano i giochi con un'altra soluzione di server di gioco. Ciò include l'hosting GameLift autogestito di Amazon con FleetIQ e l'hosting locale o i tipi di dati peer-to-peer primitivi di cloud computing. FlexMatch i costi autonomi si basano sulla potenza di calcolo utilizzata.

Se hai o prevedi di avere più di un matchmaker in diverse AWS regioni, crea una stima per ogni regione.

 Note

Amazon GameLift FlexMatch è disponibile nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Seoul), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte), Europa (Irlanda).

Per iniziare, apri la [GameLift pagina Configura Amazon](#) di AWS Pricing Calculator. Aggiungi una descrizione, scegli una regione, quindi scegli Estimate Amazon GameLift Standalone. FlexMatch In Amazon GameLift FlexMatch, completa i seguenti campi:

- Numero massimo di giocatori simultanei (CCU di picco)

Questo è il numero massimo di giocatori che possono connettersi ai tuoi server di gioco contemporaneamente e richiedere il matchmaking. Inserisci il numero massimo giornaliero di giocatori che prevedi di abbinare nelle sessioni di gioco nella regione prescelta.

Ad esempio, se vuoi abbinare fino a 1.000 giocatori alla volta, 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. Questo valore aiuta a stimare il volume delle tue richieste di matchmaking. Se non sei sicuro del valore percentuale da utilizzare, mantieni il valore predefinito di **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 tuo gioco ha una CCU oraria media di 6.000 e una CCU di picco di 10.000, inserisci il valore della percentuale. **60**

- Numero di giocatori per partita

Si tratta del numero medio di giocatori che partecipano a una sessione di gioco, come definito nel design del gioco. Se hai modalità di gioco con un numero diverso di giocatori, stima il numero medio di giocatori per sessione di gioco nell'intero gioco. Il valore di default è **8**.

- Durata del gioco (in minuti)

Questo è il periodo di tempo medio in cui i giocatori rimangono in una sessione di gioco dall'inizio alla fine. Questo valore aiuta a determinare la frequenza con cui i giocatori potrebbero aver bisogno di una nuova partita. Inserisci una durata media di gioco in minuti per i tuoi giocatori. Il valore di default è **1**.

- Complessità delle regole di matchmaking

La complessità delle regole del matchmaking si riferisce al numero e al tipo di regole che usi per abbinare i giocatori. Il livello di complessità del set di regole aiuta a determinare la quantità di potenza di calcolo richiesta per ogni partita.

- Minore complessità: scegli questa opzione se il tuo set di regole di matchmaking include poche regole, utilizza tipi di regole più semplici (come le regole di confronto) e ha regole che consentono di ottenere partite di successo con meno tentativi.
- Maggiore complessità: scegli questa opzione se il tuo set di regole di matchmaking include più regole, utilizza tipi di regole più complessi (come regole di distanza o latenza) e ha regole restrittive che comportano più errori e richiedono più tentativi di abbinamento.

Per ulteriori informazioni sulla complessità delle regole e sui prezzi, consulta [Amazon GameLift FlexMatch](#) nella pagina GameLift dei prezzi di Amazon.

Quando hai finito di configurare la stima GameLift FlexMatch dei prezzi di Amazon, scegli Aggiungi alla mia stima. Per ulteriori informazioni sulla creazione e la gestione delle stime in AWS Pricing Calculator, consulta [Creare un preventivo, configurare un servizio e aggiungere altri servizi](#) nella Guida per l'AWS Pricing Calculator.

Quote e regioni supportate

Per le quote dei GameLift servizi AWS Amazon, consulta [GameLiftQuote Amazon](#).

Per informazioni sulla richiesta di aumenti delle quote per AWS le risorse, vedere [AWSservice lquotas](#).

Per un elenco delle regioni Amazon Regioni AWS che supportanoGameLift, consulta le [GameLiftRegioni Amazon](#).

- [SDK per client in tempo reale](#)
- [Plugin per Unity](#)
- [Plugin per Unreal](#)
- [Successivo](#)

Versioni precedenti

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
14/12/2023	1.11.225 o successivo	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-11-02	1.11.193 o successivo	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-09-28	1.11.144 o successivo	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2023-08-17	1.11.144 o successivo	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-07-27	1.11.111 o successivo	5.1.0	5.1.0	5.0.2	5.0.0	1.2.0
2023-06-29	1.11.111 o successivo		5.0.4	5.0.2	5.0.0	1.2.0
2023-06-15	1.11.87 o versione successiva		5.0.4	5.0.2	5.0.0	1.2.0
2023-05-25	1.11.87 o versione successiva		5.0.3	5.0.2	5.0.0	1.2.0

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2023-04-20	1.11.63 o successivo		3.4.2		5,0,0	1.2.0
2023-04-13	1.10.21 o successivo		3.4.2	5,0,0	5,0,0	1.2.0
2023-02-09	1.10.21 o successivo		3.4.2	3.4.0	5,0,0	1.2.0
2023-01-31	1.10.21 o successivo		5,0,0	3.4.0	5,0,0	1.2.0
2022-12-01	1.10.21 o successivo		5,0,0	3.4.0		1.2.0
2022-08-25	1.9.333 o versione successiva		3.4.2	3.4.0		1.2.0
2021-10-28	1.9.133 o versione successiva		3.4.2	3.4.0		1.2.0
2021-06-03	1.8.168 o successivo		3.4.2	3.4.0		1.2.0
2021-03-23	1.8.168 o successivo		3.4.1	3.3.3		1.1.0

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2021-03-16	1.8.163 o successivo		3.4.1	33.3		1.1.0
2021-02-09	1.8.139 o successivo		3.4.1	33.3		1.1.0
2020-12-22	1.8.95 o successivo		3.4.1	33.3		1.1.0
2020-11-24	1.8.95 o successivo		3.4.1	3.3.2		1.1.0
2020-11-11	1.8.36 o successivo		3.4.1	3.3.2		1.1.0
2020-09-17	1.8.36 o successivo		3.4.1	3.3.2		1.1.0
2020-08-27	1.7.310 o versione successiva		3.4.0	3.3.1		1.1.0
2020-04-16	1.7.310 o versione successiva		3.4.0	3.3.1		1.1.0
2020-04-02	1.7.310 o versione successiva		3.4.0			1.1.0

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2019-12-19	1.7.249 o versione successiva		3.4.0			1.1.0
2019-11-14	1.7.210 o versione successiva		3.4.0			1.1.0
24-10-2019	1.7.210 o versione successiva		3.4.0			1.1.0
03-09-2019	1.7.175 o versione successiva		3.4.0			1.1.0
09-07-2019	1.7.140 o versione successiva		3.3.0			1.0.0
2019-04-25	1.7.91 o versione successiva		3.3.0			1.0.0
2019-03-07	1.7.65 o versione successiva		3.3.0			

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2019-02-07	1.7.45 o versione successiva		3.3.0			
2018-12-14	1.6.20 o versione successiva		3.3.0			
2018-09-27	1.6.20 o versione successiva		3.2.1			
2018-06-14	1.4.47 o versione successiva		3.2.1			
2018-05-10	1.4.47 o versione successiva		3.2.1			
2018-02-15	1.3.58 o versione successiva		3.2.1			
2018-02-08	1.3.52 o versione successiva		3.2.0			

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2017-09-01	1.1.43 o versione successiva		3.1.7			
2017-08-16	1.1.31 o versione successiva		3.1.7			
2017-05-16	1.0.122 o versione successiva		3.1.5			
2017-04-11	1.0.103 o versione successiva		3.1.5			
2017-02-21	1.0.72 o versione successiva		3.1.5			
2016-11-18	1.0.31 o versione successiva		3.1.0			
2016-10-13	1.0.17 o versione successiva		3.1.0			

Rilascio di servizio	AWS SDK	SDK per server				SDK per client in tempo reale
		Plugin C# per Unity	C++	Plugin C++ per Unreal	Go	
2016-09-01	0.14.9 o versione successiva		3.1.0			
2016-08-04	0.12.16 o versione successiva		3,07			

Note di rilascio

Le seguenti note di rilascio sono in ordine cronologico, con gli ultimi aggiornamenti elencati per primi. Amazon GameLift 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 aprile 2024: Amazon GameLift lancia flotte di container

Amazon offre ora GameLift un'anteprima delle flotte di container, che offrono portabilità, scalabilità, tolleranza ai guasti e agilità migliorate.

Nelle flotte di container, le istanze Amazon EC2 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 SDK e pacchetti software. Dopo aver caricato il container nel tuo Amazon Elastic Container Registry privato, Amazon GameLift 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 GameLift contenitori Amazon](#), Amazon GameLift Developer Guide
- [CreateContainerGroupDefinition](#), Riferimento alle GameLift API Amazon

13 febbraio 2024: Amazon GameLift lancia miglioramenti agli SDK e semplifica l'installazione del plug-in Amazon GameLift per 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 flotte EC2 GameLift gestite da Amazon leggono i parametri del server direttamente

dalle variabili di ambiente. I server di gioco sulle GameLift Anywhere flotte Amazon devono effettuare chiamate `InitSDK()` con i parametri del server.

Versioni aggiornate dei plugin:

- GameLift Plugin Amazon per Unreal Engine, versione 1.1.0
- GameLift Plugin Amazon 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:

- [GameLift Plugin Amazon per Unreal Engine] Ha aggiornato le istruzioni di installazione e ha semplificato la confezione. Questo plugin ora include l'ultima versione di C++ Server SDK per Unreal.
- I plugin sono stati aggiornati per supportare l'ultima versione di Server SDK. GameLift

Ulteriori informazioni:

- [Integrazione di giochi con il GameLift plug-in Amazon per Unreal Engine](#), Amazon GameLift Developer Guide
- [Download di GameLift plugin e SDK per Amazon](#)

14 dicembre 2023: Amazon GameLift 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 proprietà specifiche nelle sessioni di gioco. 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 tuo 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 a un client di gioco”](#), Guida per GameLift sviluppatori Amazon
- [GameProperty](#), Riferimento alle GameLift API Amazon
- [UpdateGameSession](#), Riferimento alle GameLift API Amazon
- [SearchGameSessions](#), Riferimento alle GameLift API Amazon

21 novembre 2023: Amazon GameLift 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 GameLift risorse Amazon utilizzando gli strumenti Infrastructure as Code (IaC). Questi strumenti includono AWS CloudFormation e anche strumenti di terze parti come Terraform e Pulumi. Grazie a 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 GameLift risorse Amazon utilizzando l'API AWS Cloud Control. Puoi continuare a lavorare con le risorse utilizzando le GameLift API di Amazon o i AWS CloudFormation modelli per Amazon GameLift.

Per dettagli sulle GameLift risorse Amazon disponibili tramite IaC, consulta il [riferimento GameLift sul tipo di](#) GameLift risorsa Amazon Amazon.

Inoltre, ora puoi ridimensionare automaticamente le tue flotte utilizzando 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 API per creare, leggere, aggiornare, eliminare ed elencare 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 aggiorna il plug-in standalone per Unity

Versioni SDK aggiornate: GameLift plugin Amazon per Unity, versione 2.0.0

Il GameLift plug-in Amazon per Unity fornisce strumenti e flussi di lavoro che semplificano i passaggi per rendere operativo il gioco Unity per l'hosting cloud con Amazon. GameLift Amazon GameLift è 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 più recenti GameLift funzionalità di Amazon, tra cui la versione 5.x dell'SDK del server e il supporto per i test locali con Amazon GameLift Anywhere. Il plug-in 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:
 - Testa la tua integrazione di gioco con Amazon GameLift utilizzando la tua workstation locale come host. Questo flusso di lavoro ti aiuta a configurare una GameLift Anywhere flotta Amazon per il tuo computer locale, avviare istanze del tuo server e client di gioco, richiedere una sessione di gioco tramite Amazon GameLift e partecipare al gioco.
 - Implementa una soluzione di hosting cloud per il tuo server di gioco integrato con EC2 GameLift gestito da Amazon e risorse di supporto AWS . Questo flusso di lavoro ti aiuta a configurare il gioco per l'hosting su cloud e offre tre scenari di distribuzione:
 - 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 che si collegano 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 GameLift integrazione e distribuzione di Amazon, tra cui:

- Ogni soluzione di hosting include AWS risorse di supporto, tra cui un pool di utenti Amazon Cognito che fornisce ID giocatore unici e convalida dei giocatori. 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 Anywhere flusso di lavoro, il plug-in automatizza le impostazioni dei parametri del server richieste.
- Per il flusso di lavoro di Amazon EC2, 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 GameLift servizio Amazon e gestisce tutte le chiamate dirette al servizio Amazon GameLift.
- 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 distribuzione, comprese Anywhere le flotte di Amazon EC2, utilizzano AWS CloudFormation modelli per descrivere e distribuire AWS le risorse per la soluzione del gioco. Questi modelli sono inclusi nel download del GameLift plugin Amazon. Puoi usarli così come sono o personalizzarli per il tuo gioco.

Ulteriori informazioni:

- [Guida al GameLift plug-in Amazon per Unity per server SDK 5.x](#), Guida per GameLift sviluppatori Amazon
- [Scarica il plugin da GitHub](#)
- [Informazioni sull' GameLift hosting Amazon](#)
- [GameLift Forum Amazon](#)

2 novembre 2023: Amazon GameLift aggiunge il supporto per le credenziali condivise

Versioni SDK aggiornate: SDK 1.11.193 AWS

La nuova funzionalità di credenziali GameLift condivise di Amazon consente alle applicazioni distribuite su flotte EC2 gestite di interagire con altre risorse. AWS Questo aggiornamento riguarda le applicazioni che raggruppi e distribuisce insieme ai file binari dei server di gioco integrati con la versione 5.x o successiva dell'SDK per 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 i parametri delle istanze EC2 e altri dati, l'agente deve essere autorizzato a interagire con le tue risorse. CloudWatch 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 distribuisce la build del tuo server di gioco su ogni istanza EC2, genera un file di credenziali condiviso e lo archivia sull'istanza. Tutte le applicazioni sull'istanza possono utilizzare le credenziali condivise. Amazon aggiorna GameLift automaticamente le credenziali temporanee per tutta la durata dell'istanza.

Puoi abilitare le credenziali condivise quando crei una flotta EC2 gestita utilizzando i seguenti metodi:

- Nel flusso di lavoro di creazione della flotta di GameLift console Amazon.
- Quando si chiama l'operazione dell'API del GameLift servizio Amazon 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](#), Amazon GameLift Developer Guide
- [CreateFleet, InstanceRoleCredentialsProvider](#), Riferimento alle GameLift API Amazon
- [Configura un ruolo di servizio IAM](#), Amazon GameLift Developer Guide

28 settembre 2023: Amazon GameLift rilascia un nuovo plug-in standalone per Unreal Engine

Versioni SDK aggiornate: GameLift plugin Amazon per Unreal Engine versione 1.0.0

Il GameLift plug-in Amazon per Unreal Engine fornisce strumenti e flussi di lavoro che semplificano i passaggi per avviare e far funzionare un gioco con Amazon GameLift per il cloud hosting. Amazon GameLift è 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:

- Testa la tua integrazione di gioco con Amazon GameLift utilizzando la tua workstation locale come host. Questo flusso di lavoro ti aiuta a configurare una GameLift Anywhere flotta Amazon per il tuo computer locale, avviare istanze del tuo server e client di gioco, richiedere una sessione di gioco tramite Amazon GameLift e ottenere informazioni di connessione per la nuova sessione di gioco.
- Implementa una soluzione di hosting cloud Amazon EC2 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 GameLift risorse Amazon e risorse di supporto AWS .
- Possibilità di configurare profili utente che si collegano a un utente dell' AWS account 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 GameLift integrazione e distribuzione di Amazon, tra cui:
 - Ogni soluzione di hosting include AWS risorse di supporto, tra cui un pool di utenti Amazon Cognito di base che fornisce ID giocatore unici, un bucket Amazon S3 per lo storage, notifiche di eventi Amazon SNS e funzioni. AWS Lambda
 - Per il Anywhere flusso di lavoro, il plug-in automatizza le impostazioni dei parametri del server richieste utilizzando argomenti della riga di comando.
 - Per il flusso di lavoro di Amazon EC2, 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 GameLift servizio Amazon.
- 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, comprese Anywhere le flotte di Amazon EC2, utilizzano AWS CloudFormation modelli per descrivere le soluzioni. Il plug-in utilizza questi modelli per distribuire GameLift risorse Amazon per il tuo gioco. Questi modelli sono inclusi nel download del GameLift plug-in Amazon e sono modificabili. Puoi usarli così come sono o modificarli per il tuo gioco.

Ulteriori informazioni:

- [Integrazione di giochi con il GameLift plug-in Amazon per Unreal Engine](#), Guida per GameLift sviluppatori Amazon
- [Scarica il plugin da GitHub](#)
- [Informazioni sull' GameLift hosting Amazon](#)
- [GameLift Forum Amazon](#)

17 agosto 2023: Amazon GameLift offre hosting di server di gioco con processori AWS Graviton

Versioni SDK aggiornate: SDK 1.11.144 AWS

Con Amazon ora GameLift puoi ospitare i tuoi giochi nel cloud utilizzando istanze EC2 con processori AWS Graviton. Progettate AWS con processori basati su ARM64, le istanze Graviton offrono il miglior rapporto prezzo/prestazioni per i carichi di lavoro cloud che utilizzano 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.

Con Amazon GameLift, ora puoi scegliere tra queste nuove istanze della famiglia AWS Graviton:

- 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 EC2 basate su Graviton.
- [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 GameLift server Amazon basato 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).

27 luglio 2023: Amazon GameLift 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 di Amazon GameLift Server SDK offre aggiornamenti per C++, C# e il plug-in Unreal, oltre a un nuovo plug-in da utilizzare con il motore di gioco Unity. Gli sviluppatori di giochi integrano l'Amazon GameLift Server SDK nei server di gioco che distribuiscono per l'hosting su Amazon. GameLift

L'ultima versione dell'SDK del server contiene i seguenti aggiornamenti, che includono una serie di richieste dei clienti:

- Scarica pacchetti SDK per lingua: il [sito di GameLift download Amazon](#) aggiornato contiene pacchetti SDK per ogni lingua. Puoi scaricare le versioni attuali 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 flotte EC2 GameLift gestite da Amazon leggono i parametri del server direttamente dalle variabili di ambiente. I server di gioco sulle GameLift Anywhere flotte Amazon 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.
 - [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)
 - [Riferimento Amazon GameLift Server SDK 5.x per C++](#)
 - [Riferimento SDK 5.x per il server Amazon GameLift Unreal Engine](#)
 - Nuove versioni delle guide all'integrazione dell'SDK 5 del server per i plugin Unity e Unreal
 - [Integra Amazon GameLift in un progetto Unity](#)
 - [Integra Amazon GameLift in un progetto Unreal Engine](#)
- Aggiornamenti della documentazione aggiuntivi
 - Documentazione rivista per le operazioni delle API di GameLift servizio Amazon [GetComputeAccess](#) e [GetInstanceAccess](#) per chiarire le procedure di accesso remoto in base alla versione Amazon GameLift Server SDK in uso.
 - Descrizioni riviste per [GameSessionPlacement](#) documentare come le informazioni sulla sessione di gioco siano transitorie quando un posizionamento è in stato «in sospeso».

13 luglio 2023: Amazon GameLift aggiunge i parametri hardware della flotta

Ora puoi tenere traccia dei parametri delle prestazioni hardware per le tue flotte EC2 GameLift gestite da Amazon. Le metriche includono le metriche delle istanze EC2 per l'utilizzo della CPU, il volume del traffico di rete e l'attività di lettura/scrittura del disco. Per Amazon GameLift, queste metriche descrivono tutte le istanze attive in una sede del parco veicoli. Puoi visualizzare queste metriche hardware della flotta utilizzando una CloudWatch dashboard di Amazon nel AWS Management Console. Puoi anche visualizzarli nella GameLift console Amazon nei dettagli della flotta.

Ulteriori informazioni:

- [Monitora Amazon GameLift con Amazon CloudWatch](#) (Metriche per le flotte), Amazon Developer Guide GameLift

29 giugno 2023: Amazon GameLift lancia il supporto per Amazon Linux 2023

Versioni SDK aggiornate: SDK 1.11.111 AWS

GameLift I clienti Amazon possono ora utilizzare il sistema operativo Amazon Linux 2023 per ospitare i propri server di gioco. AL2023 offre diversi miglioramenti rispetto ad AL2, inclusa la sicurezza. Questo sistema operativo è disponibile in tutti i Regioni AWS paesi ad eccezione delle regioni cinesi.

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 2025.

Ulteriori informazioni:

- [Domande frequenti GameLift su Amazon Linux Server](#)
- [Confronto tra Amazon Linux 2 e Amazon Linux 2023](#)
- Link di riferimento per le GameLift API Amazon:
 - [AWS Azione SDK CreateBuild](#)
 - [Comando CLI upload-build](#)
 - [Comando CLI create-build](#)

25 maggio 2023: Amazon GameLift FleetiQ 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 utilizzi Amazon GameLift FleetiQ per l'hosting di giochi, ora puoi impedire il posizionamento delle sessioni di gioco su istanze che al momento si stanno esaurendo. 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 FleetiQ esamina solo le istanze attive durante la ricerca e la rivendicazione di un server di gioco disponibile.

Ulteriori informazioni:

- [ClaimGameServer](#) nell'Amazon GameLift API Reference
- [Come funziona FleetiQ nella Amazon FleetiQ Developer Guide GameLift](#)

16 maggio 2023: Amazon GameLift supporta l'etichettatura dell'allocazione dei costi per le flotte

GameLift I clienti Amazon possono ora utilizzare i tag di allocazione dei AWS Billing costi per organizzare i costi di hosting dei giochi. Puoi assegnare tag di allocazione dei costi a singole risorse della flotta Amazon GameLift EC2 per tenere traccia del contributo delle tue flotte ai costi complessivi di hosting.

Ulteriori informazioni:

- [Gestisci i costi di hosting dei giochi](#)
- [Utilizzo dei tag di allocazione AWS dei costi](#), Guida per l'utente AWS Billing

20 aprile 2023: Amazon GameLift lancia il supporto per Windows Server 2016

Versioni SDK aggiornate: SDK 1.11.63 AWS

GameLift I clienti Amazon 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 SDK o CLI, ora devi impostare in modo esplicito 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 GameLift console Amazon, 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:

- Link di riferimento per le GameLift API Amazon:
 - [Comando CLI upload-build](#)
 - [Comando CLI create-build](#)
 - [AWS Azione SDK CreateBuild](#)
- [GameLift Domande frequenti su Amazon per Windows 2012](#)

13 aprile 2023: Amazon GameLift lancia il server SDK 5.x per Unreal

Versioni SDK aggiornate: Server SDK 5.0.0 per Unreal

L'ultima versione del plug-in GameLift leggero Amazon per Unreal Engine è ora basata sull'SDK 5.x del GameLift server Amazon. Per iniziare a integrare il tuo ambiente Unreal Engine con Amazon, GameLift consulta i seguenti link.

Ulteriori informazioni:

- [Integra Amazon GameLift in un progetto Unreal Engine](#)
- [Aggiungi Amazon GameLift al tuo server di gioco](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C++](#)

14 marzo 2023: Amazon GameLift lancia una nuova esperienza di console

La nuova GameLift console Amazon include questi miglioramenti:

- Navigazione migliorata: il nuovo pannello di navigazione facilita la navigazione tra GameLift le risorse di Amazon.
- Pagina iniziale di Amazon GameLift : la nuova pagina di destinazione fornisce collegamenti a documentazione utile, mostra una panoramica di alto livello di Amazon GameLift e fornisce supporto tramite collegamenti alla documentazione, domande frequenti e AWS re:Post.
- Parametri Amazon CloudWatch migliorati: i GameLift parametri di Amazon sono ora disponibili sia nella GameLift console Amazon che nelle dashboard. CloudWatch Questo aggiornamento include anche nuove metriche per prestazioni, utilizzo e sessioni di gioco.

Ulteriori informazioni:

- [Visualizzazione dei dati di gioco nella console](#)
- [Gestione delle risorse GameLift di hosting Amazon](#)
- [Costruire un matchmaker FlexMatch](#)

14 febbraio 2023: Amazon GameLift 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: Amazon GameLift 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 l'Amazon GameLift Server SDK 5.0.0 con .NET 4.6. Unity non supporta .NET 6.

Ulteriori informazioni:

- Scarica la versione più recente dell'SDK per GameLift server Amazon su [Amazon per GameLift iniziare](#)
- [Riferimento Amazon GameLift Server SDK 5.x per C# e Unity](#)

31 gennaio 2023: l'SDK GameLift del server Amazon supporta la lingua Go

Versioni SDK aggiornate: Server SDK 5.0.0 per Go

Ulteriori informazioni:

- Scarica la versione più recente dell'SDK per GameLift server Amazon su [Amazon per GameLift iniziare](#)
- [Riferimento SDK GameLift del server Amazon per Go](#)

1 dicembre 2022: Amazon GameLift lancia Amazon GameLift Anywhere e Amazon GameLift Server SDK 5.0

Versioni SDK aggiornate: SDK 1.10.21, Server AWS SDK 5.0.0 per C++ e C#

Amazon GameLift Anywhere utilizza le risorse del tuo server di gioco per ospitare server di GameLift gioco Amazon. Puoi usare Amazon GameLift Anywhere per integrare le tue risorse di elaborazione con l'elaborazione EC2 GameLift gestita da Amazon per distribuire i tuoi server di gioco su più tipi di elaborazione. Puoi anche utilizzare Amazon GameLift Anywhere per testare in modo iterativo i tuoi server di gioco senza caricare la build su Amazon GameLift per ogni iterazione.

Punti salienti:

- Nuovi tipi di GameLift Anywhere flotta e calcolo Amazon
- Registrazione delle risorse di GameLift Anywhere calcolo Amazon
- Ciclo di iterazione dei test migliorato

Amazon GameLift Server 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 Anywhere e l'uso delle tue risorse di elaborazione per l'hosting di server di gioco.

Ulteriori informazioni:

- [Riferimento SDK per GameLift server Amazon](#)
- [Ubicazione della flotta](#)
- [Scelta delle GameLift risorse di elaborazione Amazon](#)
- [Crea una GameLift Anywhere flotta Amazon](#)

25 agosto 2022: Amazon GameLift lancia il supporto per Local Zones

Versioni SDK aggiornate: SDK 1.9.333 AWS

Amazon GameLift è ora disponibile in otto Local Zones negli Stati Uniti, quindi puoi schierare le tue flotte più vicino ai giocatori. Puoi utilizzare tutte le GameLift funzionalità gestite di Amazon 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:

- [Zone locali](#)
- [Ubicazione della flotta](#)
- [Crea una flotta GameLift gestita da Amazon](#)

28 giugno 2022: Amazon GameLift lancia una nuova esperienza di console opt-in

La nuova GameLift console Amazon include questi miglioramenti:

- Navigazione migliorata: il nuovo pannello di navigazione facilita la navigazione tra GameLift le risorse di Amazon.
- Pagina iniziale di Amazon GameLift : la nuova pagina di destinazione fornisce collegamenti a documentazione utile, mostra una panoramica di alto livello di Amazon GameLift e fornisce supporto tramite collegamenti alla documentazione, domande frequenti e AWS re:Post.
- Parametri Amazon CloudWatch migliorati: i GameLift parametri di Amazon sono ora disponibili sia nella GameLift console Amazon che nelle dashboard. CloudWatch Questo aggiornamento include anche nuove metriche per prestazioni, utilizzo e sessioni di gioco.

Ulteriori informazioni:

- [Visualizzazione dei dati di gioco nella console](#)
- [Gestione delle risorse GameLift di hosting Amazon](#)
- [Costruire un matchmaker FlexMatch](#)

15 febbraio 2022: FlexMatch aggiunge una regola composta e ulteriori miglioramenti

FlexMatch gli 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`, `not`, `or`.
- **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:

- [Guida per GameLift FlexMatch sviluppatori Amazon:](#)
 - [FlexMatch tipi di regole](#)
 - [FlexMatch espressioni di proprietà](#)
- [AttributeValue: SL](#)

28 ottobre 2021: Amazon GameLift aggiunge il supporto per le flotte multiregionali nella regione Asia Pacifico (Osaka); Amazon FLEETIQ aggiunge il supporto per i processori GameLift Graviton2 AWS

[Versioni](#) SDK aggiornate AWS : SDK 1.9.133

Amazon GameLift è ora disponibile nella regione Asia Pacifico (Osaka). 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 è ora disponibile nella regione Asia Pacifico (Osaka).
- GameLift I gruppi di server di gioco Amazon FLEETIQ possono ora essere configurati per gestire le famiglie di istanze Graviton2 c6g, m6g e r6g.

Ulteriori informazioni:

- [Flotta Amazon GameLift multiregionale](#)
- [CreateGameServerGroup](#)
- [AWS processore gravitonico](#)

20 settembre 2021: Amazon GameLift rilascia il plug-in per Unity

Il GameLift plug-in Amazon per Unity versione 1.0.0 contiene librerie e interfaccia utente nativa che semplificano l'accesso alle GameLift risorse Amazon e l'integrazione di Amazon GameLift nel tuo gioco Unity. Puoi utilizzare il GameLift plug-in Amazon per Unity per accedere alle GameLift API di Amazon 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 utilizzare Amazon GameLift Local per vedere i messaggi trasmessi tra il client di gioco e il server di gioco per scoprire come un gioco tipico interagisce con Amazon GameLift.

Il plug-in 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:

- [Integrazione di giochi con il GameLift plug-in Amazon 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 in base all'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 per client e server di Amazon in GameLift 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 IL2CPP nelle tue applicazioni mobili che utilizzano RTS Client SDK e seguire le migliori pratiche con i framework. Ora puoi anche creare l'Amazon GameLift Server SDK per Unreal versione 4.26. Questo aggiornamento contiene componenti che si integrano con il tuo server di gioco Windows o Linux, incluse le versioni C++ e C# di Amazon GameLift Server SDK, Amazon GameLift Local e un plug-in Unreal Engine.

Punti salienti:

- È stato aggiunto il supporto per IL2CPP 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:

- [Guida per GameLift sviluppatori Amazon:](#)
 - [DescribePlayerSessions\(\)](#)
 - [GetInstanceCertificate\(\)](#)

23 marzo 2021: Amazon GameLift 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:

- Guida per GameLift sviluppatori Amazon:
 - [Imposta la notifica degli eventi per il posizionamento della sessione di gioco](#) (nuovo)
 - [Eventi di collocamento delle sessioni di gioco](#) (nuovo)
- [Riferimento API \(AWS SDK\)](#)
 - Nuovi parametri di coda delle sessioni di gioco `NotificationTarget` e `CustomEventData: GameSessionQueue`, [CreateGameSessionQueueUpdateGameSessionQueue](#)
- [GameLiftForum Amazon](#)

16 marzo 2021: Amazon GameLift aggiunge flotte multiregionali, sei nuove regioni

Versioni SDK aggiornate: [SDK 1.8.163 AWS](#)

L'hosting GameLift gestito di Amazon è ora disponibile in 21 AWS regioni. Le nuove regioni sono Città del Capo (af-south-1), Bahrein (me-south-1), Hong Kong (ap-east-1), Milano (eu-south-1), Parigi (eu-west-3) e Stoccolma (eu-north-1).

Con la nuova funzionalità delle flotte GameLift multisede di Amazon, ora puoi configurare un'unica flotta per ospitare i tuoi server di gioco in una o tutte le 20 regioni GameLift supportate da Amazon (esclusa la regione di Pechino). Questa funzionalità mira a ridurre in modo significativo il lavoro

necessario per configurare e gestire le risorse di GameLift hosting Amazon 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 ap-northeast-2 (Seoul). 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. GameLiftI prezzi di Amazon si basano sul tipo, l'ubicazione e il volume delle istanze utilizzate. (Per ulteriori informazioni, consulta [GameLiftI prezzi di Amazon](#)). AWS CloudFormation il supporto per flotte con più sedi sarà presto disponibile.

Note

Le flotte con più sedi non sono disponibili nelle regioni cinesi. GameLiftLe risorse Amazon che risiedono nelle regioni della Cina non possono interagire o essere utilizzate da risorse in altre GameLift regioni Amazon.

Punti salienti:

- Con una flotta con più sedi, aggiungi esplicitamente un elenco di località remote. Amazon GameLift distribuisce 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 i parametri di hosting nella GameLift console Amazon, aggregati per tutte le sedi di una flotta o suddivisi per ciascuna sede 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](#),, [DescribeEC2UpdateFleetCapacityInstanceLimits](#),,, [DescribeInstancesStopFleetActionsStartFleetActions](#)
- Operazioni di posizionamento delle sessioni di gioco aggiornate, con nuove funzionalità di priorità e filtro:,,
[CreateGameSessionQueueDescribeGameSessionQueuesUpdateGameSessionQueue](#)
- Operazioni di creazione di sessioni di gioco aggiornate, con nuovo supporto di localizzazione: [CreateGameSession](#),, [DescribeGameSessions](#),
[DescribeGameSessionDetailsSearchGameSessions](#)
- [Guida per GameLift sviluppatori Amazon](#):
 - [Sedi GameLift di hosting Amazon](#)(aggiornato)
 - [Guida alla progettazione del GameLift parco veicoli Amazon](#) (nuovo)

[Scalabilità della capacità di GameLift hosting di Amazon](#)(aggiornato)
 - [Progetta una coda per le sessioni di gioco](#) (nuovo)
 - [Visualizza i dettagli della flotta](#)(aggiornato)
- [GameLiftForum Amazon](#)

9 febbraio 2021: Amazon GameLift estende il supporto per le istanze AMD, autonome FlexMatch

[Versioni SDK aggiornate: SDK 1.8.139 AWS](#)

Questa versione include i seguenti aggiornamenti:

- GameLift I gruppi di server di gioco Amazon FLEETIQ possono ora essere configurati per gestire le famiglie di istanze AMD C5a, M5a e R5a. I tipi di istanze Amazon EC2 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 FleetiQ non sono attualmente disponibili per l'uso nella regione Cina (Pechino). AWS Vedi [Disponibilità delle funzionalità e differenze di implementazione](#) in Cina.

- L'hosting di giochi GameLift gestito da Amazon 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 istanze EC2 supportati, elencati per parco istanze [InstanceType](#), ora includono quanto segue:
 - 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 FlexMatch può ora 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 sull' FlexMatchhosting GameLift gestito da Amazon o con una soluzione di hosting non Amazon GameLift, consulta la [Amazon GameLift FlexMatch Developer Guide](#).
- Quando configuri le notifiche di eventi per Amazon GameLift FlexMatch, ora puoi designare un argomento Amazon SNS FIFO come destinazione della notifica. Per ulteriori informazioni, consultare:
 - [MatchmakingConfiguration NotificationTarget](#), Riferimento alle GameLift API Amazon
 - [Configurare la notifica FlexMatch degli eventi](#), Amazon GameLift FlexMatch Developer Guide
 - [Presentazione della messaggistica Amazon SNS FIFO — F irst-in-first-out Pub/Sub](#), notizie e blog AWS

22 dicembre 2020: l'SDK GameLift del server Amazon supporta Unreal Engine 4.25 e Unity 2020

Versioni SDK aggiornate: Amazon GameLift Server SDK 4.0.2, plugin Unreal versione 3.3.3

L'ultima versione di Amazon GameLift 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 l'ultima versione di Amazon GameLift Server SDK da Amazon [GameLift Getting Started](#).

24 novembre 2020: Amazon GameLift FlexMatch ora disponibile per i giochi ospitati ovunque

Versioni SDK aggiornate: AWS [SDK 1.8.95](#)

Amazon GameLift FlexMatch è un servizio di matchmaking personalizzabile per giochi multiplayer. Progettato inizialmente per gli utenti dell'hosting GameLift gestito di Amazon, ora FlexMatch può essere integrato in giochi che utilizzano altri sistemi di hosting peer-to-peer, tra cui l'elaborazione locale proprietaria e i tipi primitivi di cloud computing. I giochi che utilizzano Amazon GameLift FleetIQ per l'hosting di giochi su Amazon EC2 ora possono implementare il matchmaking con FlexMatch

FlexMatch fornisce 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 dei giocatori 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. Quando lo utilizzi FlexMatch con Amazon GameLift managed hosting o Realtime Servers, il matchmaker utilizza automaticamente Amazon GameLift 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 GameLift servizio Amazon, 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 l'hosting GameLift gestito di Amazon 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: istanze AMD ora disponibili su Amazon GameLift

[Versioni SDK aggiornate: SDK 1.8.95 AWS](#)

L'elenco dei tipi di istanze Amazon EC2 supportati da Amazon GameLift ora 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 x86; i giochi attualmente in esecuzione su Amazon GameLift 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](#)
- [Prezzi delle GameLift istanze Amazon](#)
- [Istanze Amazon EC2 con processori AMD EPYC](#)
- [GameLiftForum Amazon](#)

11 novembre 2020: aggiornamento della versione ad Amazon GameLift Server SDK

Versioni SDK aggiornate: Amazon GameLift 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 l'ultima versione di Amazon GameLift Server SDK da Amazon [GameLift Getting Started](#).

5 novembre 2020: nuove personalizzazioni degli FlexMatch algoritmi

FlexMatch gli 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 agli GameLift SDK Amazon.

- 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. Ordinando 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 in base agli 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 aggiorna l'SDK del server

Versioni SDK aggiornate: Amazon GameLift 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'operazione [GetInstanceCertificate\(\)](#).
 - 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 [F ProcessParameters](#) supportare il match backfill.

Scarica l'ultima versione di Amazon GameLift Server SDK da Amazon [GameLift Getting Started](#).

27 agosto 2020: Amazon GameLift FleetiQ per l'hosting di giochi con Amazon EC2 (disponibilità generale)

[Versioni SDK aggiornate: SDK 1.8.36AWS](#)

La soluzione Amazon GameLift FleetiQ per l'hosting di giochi a basso costo e basato sul cloud su Amazon EC2 è ora disponibile a livello generale. Amazon GameLift FleetiQ offre agli sviluppatori la possibilità di ospitare server di gioco direttamente su istanze Spot Amazon EC2 ottimizzando la loro fattibilità per l'hosting di giochi. Gli sviluppatori di giochi possono utilizzare Amazon GameLift FleetiQ con nuovi giochi o per 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 alla soluzione Amazon GameLift FleetiQ:

- La nuova operazione API `DescribeGameServerInstances` restituisce informazioni, incluso lo stato, su tutte le istanze attive per un gruppo di server di gioco Amazon GameLift FleetiQ.

- 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 chiavi di ordinamento personalizzate 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 aggiorna l'SDK del server per Unity e Unreal Engine

Versioni SDK aggiornate: Amazon GameLift Server SDK 4.0.0, Amazon Local 1.0.5 GameLift

L'ultima versione di Amazon GameLift Server SDK 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 Local versione 1.0.5 è stata aggiornata per testare le integrazioni che utilizzano la versione 4.0.0 dell'SDK per server C#.

Scarica l'ultima versione di Amazon GameLift Server SDK da Amazon [GameLift Getting Started](#).

2 aprile 2020: Amazon GameLift FLEETIQ disponibile per l'hosting di giochi su EC2 (anteprima pubblica)

[Versioni SDK aggiornate: SDK 1.7.310AWS](#)

La funzionalità Amazon GameLift FLEETIQ ottimizza la fattibilità delle istanze Spot a basso costo da utilizzare con l'hosting di giochi. Questa funzionalità è ora estesa per i clienti che desiderano gestire le proprie risorse di hosting direttamente anziché tramite il GameLift servizio Amazon 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 di blog](#) su Amazon GameLift fleetiq

19 dicembre 2019: migliore gestione AWS delle risorse per GameLift le risorse Amazon

Versioni SDK aggiornate: AWS [SDK 1.7.249](#)

Ora puoi sfruttare gli strumenti di gestione AWS delle risorse con GameLift le risorse di Amazon. In particolare, a tutte le GameLift risorse Amazon 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, GameLift le risorse di Amazon ora supportano l'etichettatura. 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 AWS costi, ecc. Quando gestisci i tag per GameLift le risorse Amazon, utilizza le azioni `TagResource()` dell' GameLift API Amazon e `ListTagsForResource()`. `UntagResource()`

Ulteriori informazioni:

- [TagResource](#) nell'Amazon GameLift API Reference
- [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 e aggiornamenti nella regione Cina (Pechino)

Versioni SDK aggiornate: AWS [SDK 1.7.210](#)

AWS CloudFormation modelli per Amazon GameLift

GameLift Le risorse Amazon 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:

- [Riferimento al tipo di GameLift risorsa Amazon](#) nella Guida AWS CloudFormation per l'utente
- [Gestisci le risorse utilizzando AWS CloudFormation](#) nella Amazon GameLift Developer Guide

Glossario per AWS

Per la terminologia AWS più recente, consultare il [glossario AWS](#) nella documentazione di riferimento per Glossario AWS.

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.