



FlexMatch Guida per gli sviluppatori

Amazon GameLift Servers



Version

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon GameLift Servers: FlexMatch Guida per gli sviluppatori

Copyright © 2025 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 cosa è FlexMatch?	1
Chiave FlexMatch caratteristiche	2
FlexMatch con Amazon GameLift Servers hosting	3
Prezzi per Amazon GameLift Servers FlexMatch	3
In che modo FlexMatch funzionamento	4
Componenti del matchmaking	4
FlexMatch processo di matchmaking	6
Supportato Regioni AWS	8
Nozioni di base	9
Configura un account per FlexMatch	9
Tabella di marcia: Crea una soluzione di matchmaking autonoma	10
Tabella di marcia: Aggiungi matchmaking a Amazon GameLift Servers hosting	12
Costruire un FlexMatch parainfo	14
Progettare un abbinamento	15
Configura un matchmaker di base	15
Scegli una posizione per il matchmaker	16
Aggiungi elementi opzionali	16
Creare un set di regole	18
Progetta un set di regole	19
Progetta un set di regole per partite di grandi dimensioni	27
Tutorial: Crea un set di regole	31
esempi di set di regole	34
Crea una configurazione di matchmaking	59
Tutorial: Crea un matchmaker per l'hosting	60
Tutorial: Crea un matchmaker indipendente FlexMatch	62
Tutorial: Modifica una configurazione di matchmaking	65
Configurare le notifiche degli eventi	65
Imposta eventi EventBridge	66
Tutorial: configurare un argomento Amazon SNS	66
Imposta un argomento SNS con crittografia lato server	68
Configurare un abbonamento a un argomento per richiamare una funzione Lambda	69
Preparazione di giochi per FlexMatch	71
Add (Aggiungi) FlexMatch a un client di gioco	71
Attività prerequisite sul lato client	72

Richiedi il matchmaking per i giocatori	73
Tieni traccia degli eventi di matchmaking	75
Richiedi l'accettazione del giocatore	75
Connect a una partita	76
Esempi di richieste di matchmaking	77
Add (Aggiungi) FlexMatch a un server di gioco	78
Informazioni sui dati del matchmaker	79
Configura un server di gioco per FlexMatch	80
Riempi i giochi esistenti	82
Attiva il riempimento automatico	82
Genera richieste di riempimento manuali da un server di gioco	83
Genera richieste di riempimento manuali da un servizio di backend	86
Aggiorna i dati delle partite sul server di gioco	89
Sicurezza con FlexMatch	91
FlexMatch riferimento	92
FlexMatch Riferimento API (AWS SDK)	92
Imposta regole e processi di matchmaking	92
Richiedi una partita per uno o più giocatori	93
Linguaggi di programmazione compatibili	94
Linguaggio delle regole	94
schema del set di regole	94
definizioni delle proprietà dei set di regole	98
tipi di regole	105
espressioni di proprietà	112
eventi di matchmaking	117
MatchmakingSearching	117
PotentialMatchCreated	118
AcceptMatch	120
AcceptMatchCompleted	122
MatchmakingSucceeded	123
MatchmakingTimedOut	125
MatchmakingCancelled	127
MatchmakingFailed	128
Note di rilascio e versioni SDK	130
Tutti Amazon GameLift Servers guide	131
.....	cxxxii

Che cos'è Amazon GameLift Servers FlexMatch?

Amazon GameLift Servers FlexMatch è un servizio di matchmaking personalizzabile per giochi multiplayer. Con FlexMatch, puoi creare un set personalizzato di regole che definiscono l'aspetto di una partita multiplayer per il tuo gioco e determinano come valutare e selezionare i giocatori compatibili per ogni partita. Puoi anche perfezionare gli aspetti chiave dell'algoritmo di matchmaking per adattarli alle tue esigenze di gioco.

Utilizzo FlexMatch come servizio di matchmaking autonomo o integrato con un Amazon GameLift Servers soluzione di hosting di giochi. Ad esempio, potresti implementare FlexMatch come funzionalità autonoma con giochi con un' peer-to-peerarchitettura o giochi che utilizzano altre soluzioni di cloud computing. Oppure potresti aggiungere FlexMatch al tuo Amazon GameLift Servers hosting di container gestiti EC2 o gestiti o hosting locale con Amazon GameLift Servers Ovunque. Questa guida fornisce informazioni dettagliate su come creare un FlexMatch sistema di matchmaking per il tuo scenario particolare.

FlexMatch ti offre la flessibilità di impostare le priorità di matchmaking in base alle tue esigenze di gioco. Ad esempio, puoi eseguire le operazioni seguenti:

- Trova un equilibrio tra velocità e qualità della partita. Imposta le regole delle partite per trovare rapidamente le partite che sono abbastanza buone, oppure chiedi ai giocatori di aspettare ancora un po' per trovare la migliore partita possibile per un'esperienza di gioco ottimale.
- Crea partite basate su giocatori o squadre ben assortite. Crea partite in cui tutti i giocatori abbiano caratteristiche simili come abilità o esperienza. Oppure organizza partite in cui le caratteristiche combinate di ciascuna squadra soddisfino criteri comuni.
- Dai la priorità al modo in cui la latenza dei giocatori influisce sul matchmaking. Vuoi impostare un limite rigido di latenza per tutti i giocatori o sono accettabili latenze più elevate purché tutti i partecipanti alla partita abbiano una latenza simile?

Pronto per iniziare a lavorare con FlexMatch?

Per step-by-step ricevere assistenza su come installare e utilizzare il gioco FlexMatch, consulta i seguenti argomenti:

- [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#)

- [Tabella di marcia: Crea una soluzione di matchmaking autonoma con FlexMatch](#)

Chiave FlexMatch caratteristiche

Le seguenti funzionalità sono disponibili con tutti FlexMatch scenari, indipendentemente dal fatto che si utilizzino FlexMatch come servizio autonomo o con Amazon GameLift Servers hosting di giochi.

- Abbinamento personalizzabile dei giocatori. Progetta e crea matchmaker adatti a tutte le modalità di gioco che offri ai tuoi giocatori. Crea una serie di regole personalizzate per valutare gli attributi chiave dei giocatori (come il livello di abilità o il ruolo) e i dati sulla latenza geografica per creare partite tra giocatori eccezionali per il tuo gioco.
- Corrispondenza basata sulla latenza. Fornisci dati sulla latenza dei giocatori e crea regole di partita che richiedono ai giocatori in una partita di avere tempi di risposta simili. Questa funzione è utile quando i tuoi pool di matchmaking di giocatori si estendono su più aree geografiche.
- Supporto per partite di dimensioni fino a 200 giocatori. Crea partite con un massimo di 40 giocatori utilizzando regole personalizzate per il tuo gioco. Crea partite con un massimo di 200 giocatori utilizzando un processo di abbinamento che utilizza un processo di abbinamento personalizzato semplificato per mantenere gestibili i tempi di attesa dei giocatori.
- Accettazione dei giocatori. Richiedi ai giocatori di iscriversi a una partita proposta prima di finalizzarla e iniziare una sessione di gioco. Usa questa funzione per avviare un flusso di lavoro di accettazione personalizzato e segnalare le risposte dei giocatori a FlexMatch prima di iniziare una nuova sessione di gioco per la partita. Se non tutti i giocatori accettano una partita, la partita proposta fallisce e i giocatori che l'hanno accettata tornano automaticamente al matchmaking pool.
- Supporto per i gruppi di giocatori. Genera partite per gruppi di giocatori che vogliono giocare insieme nella stessa squadra. Utilizzo FlexMatch per trovare altri giocatori per completare la partita, se necessario.
- Regole di abbinamento espandibili. Allenta gradualmente i requisiti della partita dopo che è trascorso un certo periodo di tempo senza trovare una partita vincente. L'espansione delle regole ti consente di decidere dove e quando allentare le regole iniziali della partita, in modo che i giocatori possano iniziare più rapidamente le partite giocabili.
- Riempimento della partita. Riempi gli slot vuoti in una sessione di gioco esistente con nuovi giocatori ben assortiti. Personalizza quando e come richiedere nuovi giocatori e usa le stesse regole di partita personalizzate per trovare altri giocatori.

FlexMatch con Amazon GameLift Servers hosting

FlexMatch offre le seguenti funzionalità aggiuntive da utilizzare con i giochi con cui offri servizi di hosting Amazon GameLift Servers. Ciò include giochi con server di gioco personalizzati o Amazon GameLift Servers In tempo reale.

- **Posizionamento della sessione di gioco.** Quando una partita viene effettuata con successo, FlexMatch richiede automaticamente un nuovo posizionamento nella sessione di gioco da Amazon GameLift Servers. I dati generati durante il matchmaking, comprese le assegnazioni di giocatori IDs e squadre, vengono forniti al server di gioco in modo che possa utilizzare tali informazioni per iniziare la sessione di gioco della partita. FlexMatch quindi restituisce le informazioni sulla connessione della sessione di gioco in modo che i client di gioco possano partecipare al gioco. Per ridurre al minimo la latenza sperimentata dai giocatori in una partita, posiziona la sessione di gioco con Amazon GameLift Servers può anche utilizzare i dati regionali sulla latenza dei giocatori, se forniti.
- **Riempimento automatico delle partite.** Con questa funzione abilitata, FlexMatch invia automaticamente una richiesta di completamento di una partita quando inizia una nuova sessione di gioco con slot per giocatori non occupati. Il tuo sistema di matchmaking avvia il processo di posizionamento della sessione di gioco con un numero minimo di giocatori, quindi riempie rapidamente gli slot rimanenti. Non puoi utilizzare il backfill automatico per sostituire i giocatori che abbandonano una sessione di gioco abbinata.

Se usi Amazon GameLift Servers FleetIQ con giochi ospitati con risorse Amazon Elastic Compute Cloud (Amazon EC2), implementa FlexMatch come servizio autonomo.

Prezzi per Amazon GameLift Servers FlexMatch

Amazon GameLift Servers costi per istanze in base alla durata di utilizzo e per la larghezza di banda in base alla quantità di dati trasferiti. Se ospiti i tuoi giochi su Amazon GameLift Servers server, FlexMatch l'utilizzo è incluso nelle tariffe per Amazon GameLift Servers. Se ospiti i tuoi giochi su un'altra soluzione server, FlexMatch l'utilizzo viene addebitato separatamente. Per un elenco completo delle spese e dei prezzi per Amazon GameLift Servers, vedi [Amazon GameLift Servers Prezzi](#).

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

In che modo Amazon GameLift Servers FlexMatch funzionamento

Questo argomento fornisce una panoramica delle Amazon GameLift Servers FlexMatch servizio, inclusi i componenti principali di un FlexMatch sistema e come interagiscono.

È possibile utilizzare... FlexMatch con giochi che usano Amazon GameLift Servers hosting gestito o con giochi che utilizzano un'altra soluzione di hosting. Giochi ospitati su Amazon GameLift Servers server, tra cui Amazon GameLift Servers In tempo reale, usa il sistema integrato Amazon GameLift Servers servizio per localizzare automaticamente i server di gioco disponibili e avviare le sessioni di gioco per le partite. Giochi che utilizzano FlexMatch come servizio autonomo, incluso Amazon GameLift Servers FleetIQ, deve coordinarsi con il sistema di hosting esistente per assegnare le risorse di hosting e avviare le sessioni di gioco per le partite.

Per una guida dettagliata sulla configurazione FlexMatch per i tuoi giochi, vedi [Nozioni di base su FlexMatch](#).

Componenti del matchmaking

A FlexMatch il sistema di matchmaking include alcuni o tutti i seguenti componenti.

Amazon GameLift Servers componenti

Questi sono Amazon GameLift Servers risorse che controllano il modo in cui FlexMatch il servizio esegue il matchmaking per il tuo gioco. Vengono creati e gestiti utilizzando Amazon GameLift Servers strumenti, tra cui la console e la AWS CLI o, in alternativa, l'utilizzo programmatico dell'SDK per AWS Amazon GameLift Servers.

- FlexMatch configurazione del matchmaking (chiamato anche matchmaker): un matchmaker è un insieme di valori di configurazione che personalizza il processo di matchmaking del gioco. Un gioco può avere più matchmaker, ciascuno configurato per diverse modalità o esperienze di gioco, in base alle esigenze. Quando il gioco invia una richiesta di matchmaking a FlexMatch, specifica quale matchmaker usare.
- FlexMatch set di regole di matchmaking: un set di regole contiene tutte le informazioni necessarie per valutare i giocatori in vista di potenziali partite e approvare o rifiutare. Il set di regole definisce la struttura della squadra di una partita, dichiara gli attributi del giocatore utilizzati per la valutazione e fornisce regole che descrivono i criteri per una partita accettabile. Le regole possono essere applicate a singoli giocatori, a squadre o all'intera partita. Ad esempio, una regola potrebbe richiedere che tutti i giocatori della partita scelgano la stessa mappa di gioco, oppure potrebbe richiedere che tutte le squadre abbiano una media di abilità dei giocatori simile.

- Amazon GameLift Servers coda delle sessioni di gioco (per FlexMatch con Amazon GameLift Servers (solo hosting gestito): una coda per le sessioni di gioco individua le risorse di hosting disponibili e avvia una nuova sessione di gioco per la partita. La configurazione della coda determina dove Amazon GameLift Servers cerca le risorse di hosting disponibili e come selezionare il miglior host disponibile per una partita.

Componenti personalizzati

I seguenti componenti comprendono le funzionalità necessarie per una soluzione completa FlexMatch sistema che devi implementare in base all'architettura del tuo gioco.

- Interfaccia giocatore per matchmaking: questa interfaccia consente ai giocatori di partecipare a una partita. Come minimo, avvia una richiesta di matchmaking tramite il componente del servizio di matchmaking client e fornisce dati specifici del giocatore, come i dati sul livello di abilità e sulla latenza, necessari per il processo di matchmaking.

Note

Come buona pratica, comunicazione con FlexMatch il servizio deve essere eseguito da un servizio di backend, non da un client di gioco.

- Servizio di matchmaking client: questo servizio invia le richieste di iscrizione dei giocatori dall'interfaccia del giocatore, genera richieste di matchmaking e le invia al FlexMatch servizio. Per le richieste in corso, monitora gli eventi di matchmaking, tiene traccia dello stato del matchmaking, e interviene secondo necessità. A seconda di come gestisci l'hosting delle sessioni di gioco nel gioco, questo servizio può restituire ai giocatori le informazioni sulla connessione della sessione di gioco. Questo componente utilizza l' AWS SDK con Amazon GameLift Servers API per comunicare con FlexMatch servizio.
- Servizio di collocamento delle partite (per FlexMatch (solo come servizio autonomo): questo componente funziona con il sistema di hosting dei giochi esistente per individuare le risorse di hosting disponibili e avviare nuove sessioni di gioco per le partite. Il componente deve ottenere i risultati del matchmaking ed estrarre le informazioni necessarie per iniziare una nuova sessione di gioco, compresi i giocatori IDs, gli attributi e le assegnazioni di squadra per tutti i giocatori della partita.

FlexMatch processo di matchmaking

Questo argomento descrive la sequenza degli eventi in uno scenario di matchmaking di base, comprese le interazioni tra i vari componenti del gioco e il FlexMatch servizio.

Passo 1: Richiedi il matchmaking per i giocatori

Un giocatore che utilizza il tuo client di gioco fa clic sul pulsante «Partecipa alla partita». Questa azione fa sì che il servizio di matchmaking del cliente invii una richiesta di matchmaking a FlexMatch. La richiesta identifica il FlexMatch matchmaker da utilizzare per soddisfare la richiesta. La richiesta include anche le informazioni sui giocatori richieste dal tuo matchmaker personalizzato, come il livello di abilità, le preferenze di gioco o i dati sulla latenza geografica. Puoi effettuare richieste di matchmaking per uno o più giocatori.

Passo 2: Aggiungi le richieste al pool di matchmaking

Quando FlexMatch riceve la richiesta di matchmaking, genera un ticket di matchmaking e lo aggiunge al pool di biglietti del matchmaker. Il ticket rimane nel pool fino a quando non viene abbinato o non viene raggiunto un limite di tempo massimo. Il tuo servizio di matchmaking per i clienti viene periodicamente informato sugli eventi di matchmaking, comprese le modifiche allo stato del ticket.

Passo 3: Costruisci una partita

Il tuo FlexMatch matchmaker esegue continuamente la seguente procedura su tutti i ticket del suo pool:

1. Il matchmaker ordina il pool in base all'età del biglietto, quindi inizia a costruire una potenziale partita partendo dal biglietto più vecchio.
2. Il matchmaker aggiunge un secondo ticket alla potenziale partita e valuta il risultato in base alle tue regole di matchmaking personalizzate. Se la potenziale partita supera la valutazione, i giocatori del ticket vengono assegnati a una squadra.
3. Il matchmaker aggiunge il ticket successivo in sequenza e ripete il processo di valutazione. Quando tutti i posti per i giocatori sono stati occupati, la partita è pronta.

Il matchmaking per partite di grandi dimensioni (da 41 a 200 giocatori) utilizza una versione modificata del processo sopra descritto in modo da poter creare partite in un lasso di tempo ragionevole. Invece di valutare ogni ticket singolarmente, il matchmaker divide un pool di biglietti preordinato in potenziali partite, quindi bilancia ogni partita in base alle caratteristiche del giocatore che hai specificato. Ad esempio, un matchmaker potrebbe preordinare i ticket in base a

località simili a bassa latenza e quindi utilizzare il bilanciamento post-partita per assicurarsi che le squadre siano abbinati in modo uniforme in base alle abilità dei giocatori.

Passo 4: Segnala i risultati del matchmaking

Quando viene trovata una partita accettabile, tutti i biglietti abbinati vengono aggiornati e viene generato un evento di matchmaking di successo per ogni biglietto abbinato.

- FlexMatch come servizio autonomo: il tuo gioco riceve i risultati delle partite in un evento di matchmaking di successo. I dati sui risultati includono un elenco di tutti i giocatori abbinati e dei loro incarichi di squadra. Se le richieste di partita contengono informazioni sulla latenza dei giocatori, i risultati suggeriscono anche una posizione geografica ottimale per la partita.
- FlexMatch con un Amazon GameLift Servers soluzione di hosting: i risultati delle partite vengono passati automaticamente a Amazon GameLift Servers coda per il posizionamento della sessione di gioco. Il matchmaker determina quale coda viene utilizzata per il posizionamento della sessione di gioco.

Passo 5: Inizia una sessione di gioco per la partita

Dopo che una partita proposta è stata formata con successo, viene avviata una nuova sessione di gioco. I tuoi server di gioco devono essere in grado di utilizzare i dati dei risultati del matchmaking, comprese le assegnazioni di giocatori IDs e squadre, quando organizzi una sessione di gioco per la partita.

- FlexMatch come servizio autonomo: il tuo servizio personalizzato di posizionamento delle partite ottiene i dati sui risultati delle partite dagli eventi di matchmaking andati a buon fine e si collega al sistema di posizionamento delle sessioni di gioco esistente per individuare una risorsa di hosting disponibile per la partita. Dopo aver trovato una risorsa di hosting, il servizio di posizionamento delle partite si coordina con il sistema di hosting esistente per iniziare una nuova sessione di gioco e acquisire informazioni sulla connessione.
- FlexMatch con un Amazon GameLift Servers soluzione di hosting: La coda della sessione di gioco individua il miglior server di gioco disponibile per la partita. A seconda di come è configurata la coda, cerca di collocare la sessione di gioco con le risorse più economiche e dove i giocatori sperimenteranno una bassa latenza (se vengono forniti dati sulla latenza del giocatore). Una volta completata con successo la sessione di gioco, Amazon GameLift Servers il servizio richiede al server di gioco di iniziare una nuova sessione di gioco, trasmettendo i risultati del matchmaking e altri dati di gioco opzionali.

Fase 6: Connettere i giocatori alla partita

Dopo l'inizio di una sessione di gioco, i giocatori si connettono alla sessione, rivendicano l'incarico di squadra e iniziano a giocare.

- FlexMatch come servizio autonomo: il gioco utilizza il sistema di gestione delle sessioni di gioco esistente per fornire ai giocatori le informazioni di connessione.
- FlexMatch con un Amazon GameLift Servers soluzione di hosting: se la sessione di gioco viene completata correttamente, FlexMatch aggiorna tutti i ticket abbinati con informazioni sulla connessione alla sessione di gioco e un ID di sessione del giocatore.

FlexMatch supportata Regioni AWS

Se stai usando FlexMatch con un Amazon GameLift Servers soluzione di hosting, puoi ospitare sessioni di gioco abbinata in qualsiasi luogo in cui offri giochi. Consulta l'elenco [completo Regioni AWS e le sedi di Amazon GameLift Servers ospitando](#).

Nozioni di base su FlexMatch

Usa le risorse di questa sezione per aiutarti a iniziare a creare un sistema di matchmaking con FlexMatch.

Argomenti

- [Configura un modulo Account AWSFlexMatch](#)
- [Tabella di marcia: Crea una soluzione di matchmaking autonoma con FlexMatch](#)
- [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#)

Configura un modulo Account AWSFlexMatch

Amazon GameLift Servers FlexMatch è un AWS servizio, ed è necessario disporre di un AWS account per utilizzare questo servizio. La creazione di un AWS account è gratuita. Per ulteriori informazioni su cosa puoi fare con un AWS account, vedi [Guida introduttiva AWS](#).

Se stai usando FlexMatch con altri Amazon GameLift Servers soluzioni, consulta i seguenti argomenti:

- [Configurazione dell'accesso per Amazon GameLift Servers ospitando](#)
- [Configurazione dell'accesso per l'hosting con Amazon GameLift Servers FleetIQ](#)

Per configurare il tuo account per Amazon GameLift Servers

1. Creare un account. Apri [Amazon Web Services](#) e scegli Accedi alla console. Segui le istruzioni per creare un nuovo account o accedere a uno esistente.
2. Configura un gruppo di utenti amministrativi. Apri la console di servizio AWS Identity and Access Management (IAM) e segui i passaggi per creare o aggiornare utenti o gruppi di utenti. IAM gestisce l'accesso ai tuoi AWS servizi e alle tue risorse. Tutti coloro che accedono al tuo FlexMatch risorse, utilizzando il Amazon GameLift Servers console o chiamando Amazon GameLift Servers APIs, deve avere un accesso esplicito. Per istruzioni dettagliate sull'uso della console (o della AWS CLI o altri strumenti) per configurare i gruppi di utenti, consulta [Creazione di utenti IAM](#).
3. Allega una politica di autorizzazioni al tuo utente o gruppo di utenti. L'accesso ai AWS servizi e alle risorse viene gestito associando una [policy IAM](#) a un utente o a un gruppo di utenti. Le

politiche di autorizzazione specificano una serie di AWS servizi e azioni a cui un utente deve avere accesso.

In Amazon GameLift Servers, è necessario creare una politica di autorizzazioni personalizzata e allegarla a ciascun utente o gruppo di utenti. Una policy è un documento JSON. Usa l'esempio seguente per creare la tua politica.

L'esempio seguente illustra una politica di autorizzazioni in linea con autorizzazioni amministrative per tutti Amazon GameLift Servers risorse e azioni. È possibile scegliere di limitare l'accesso specificando solo FlexMatch-articoli specifici.

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

Tabella di marcia: Crea una soluzione di matchmaking autonoma con FlexMatch

Questo argomento descrive il processo di integrazione completo per l'implementazione FlexMatch come servizio di matchmaking autonomo. Utilizza questa procedura se il gioco multiplayer è ospitato utilizzando peer-to-peer hardware locale configurato in modo personalizzato o altre primitive di cloud computing. Questo processo può essere utilizzato anche con Amazon GameLift Servers FleetIQ, che è una soluzione di ottimizzazione dell'hosting per i giochi ospitati su Amazon EC2. Se stai ospitando il tuo gioco utilizzando Amazon GameLift Servers hosting gestito (incluso Amazon GameLift Servers Realtime), vedi [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#).

Prima di iniziare l'integrazione, è necessario disporre di un AWS account e impostare le autorizzazioni di accesso per Amazon GameLift Servers servizio. Per informazioni dettagliate, consultare [Configura un modulo Account AWS FlexMatch](#). Tutte le attività essenziali relative alla creazione e alla gestione Amazon GameLift Servers FlexMatch i matchmaker e i set di regole possono essere eseguiti utilizzando il Amazon GameLift Servers console.

1. Crea un FlexMatch set di regole di matchmaking. Il set di regole personalizzato fornisce istruzioni complete su come costruire una partita. In esso, definisci la struttura e le dimensioni di ogni squadra. Fornisci anche una serie di requisiti che una partita deve soddisfare per essere valida, quali FlexMatch usa per includere o escludere giocatori in una partita. Questi requisiti potrebbero applicarsi ai singoli giocatori. Puoi anche personalizzare il FlexMatch algoritmo nel set di regole, ad esempio per creare partite di grandi dimensioni con un massimo di 200 giocatori. Vedi questi argomenti:
 - [Costruisci un FlexMatch set di regole](#)
 - [FlexMatch esempi di set di regole](#)
2. Imposta le notifiche per gli eventi di matchmaking. Usa le notifiche per tracciare FlexMatch attività di matchmaking, compreso lo stato delle richieste di match in sospeso. Questo è il meccanismo utilizzato per fornire i risultati di una partita proposta. Poiché le richieste di abbinamento sono asincrone, è necessario un modo per monitorare lo stato delle richieste. L'utilizzo delle notifiche è l'opzione preferita a tal fine. Vedi questi argomenti:
 - [Configurazione FlexMatch notifiche di eventi](#)
 - [FlexMatch eventi di matchmaking](#)
3. Configura un FlexMatch configurazione del matchmaking. Chiamato anche matchmaker, questo componente riceve richieste di matchmaking e le elabora. Si configura un matchmaker specificando un set di regole, un obiettivo di notifica e il tempo massimo di attesa. Puoi anche abilitare funzionalità opzionali. Vedi questi argomenti:
 - [Progetta un FlexMatch paraninfo](#)
 - [Crea una configurazione di matchmaking](#)
4. Crea un servizio di matchmaking per clienti. Crea o espandi un servizio client di gioco con funzionalità per creare e inviare richieste di matchmaking a FlexMatch. Per creare richieste di matchmaking, questo componente deve disporre di meccanismi per ottenere i dati dei giocatori richiesti dal set di regole di matchmaking e, facoltativamente, informazioni sulla latenza regionale. Deve inoltre avere un metodo per creare e assegnare un ticket unico per ogni richiesta. IDs Potresti anche scegliere di creare un flusso di lavoro di accettazione dei giocatori che richieda ai giocatori di iscriversi a una partita proposta. Questo servizio deve inoltre monitorare gli eventi di matchmaking per ottenere i risultati delle partite e avviare il posizionamento delle sessioni di gioco per partite di successo. Vedi questo argomento:
 - [Add \(Aggiungi\) FlexMatch a un client di gioco](#)

5. Crea un servizio di posizionamento delle partite. Crea un meccanismo che funzioni con il tuo sistema di hosting di giochi esistente per individuare le risorse di hosting disponibili e avviare nuove sessioni di gioco per partite di successo. Questo componente deve essere in grado di utilizzare le informazioni sui risultati delle partite per ottenere un server di gioco disponibile e iniziare una nuova sessione di gioco per la partita. Potresti anche voler implementare un flusso di lavoro per effettuare richieste di riempimento delle partite, che utilizza il matchmaking per riempire gli slot aperti nelle sessioni di gioco abbinata che sono già in corso.

Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting

FlexMatch è disponibile con il gestore Amazon GameLift Servers hosting per server di gioco personalizzati e Amazon GameLift Servers In tempo reale. Per aggiungere FlexMatch matchmaking al tuo gioco, completa le seguenti attività.

- Configurare un matchmaker. Un matchmaker riceve le richieste di abbinamento dai giocatori e le elabora. Raggruppa i giocatori in base a un set di regole definite e, per ogni abbinamento riuscito, crea nuove sessioni di gioco e nuove sessioni giocatore. Per configurare un matchmaker, segui questa procedura:
 - Creare un set di regole. Un set di regole indica al matchmaker come costruire un abbinamento valido. Specifica la struttura del team e come valutare i giocatori per l'inclusione in un abbinamento. Vedi questi argomenti:
 - [Costruisci un FlexMatch set di regole](#)
 - [FlexMatch esempi di set di regole](#)
 - Creare una coda della sessione di gioco. Una coda individua la migliore regione per ogni abbinamento e crea una nuova sessione di gioco in tale regione. Usa una coda esistente o creane una nuova per l'abbinamento. Vedi questo argomento:
 - [Crea una coda](#)
 - Configurare le notifiche (facoltativo). Poiché le richieste di abbinamento sono asincrone, è necessario un modo per monitorare lo stato delle richieste. Le notifiche sono l'opzione preferita. Vedi questo argomento:
 - [Configurazione FlexMatch notifiche di eventi](#)
 - Configurare un matchmaker. Dopo aver impostato un set di regole, una coda e la destinazione delle notifiche, creare la configurazione per il matchmaker. Vedi questi argomenti:

- [Progetta un FlexMatch parainfo](#)
- [Crea una configurazione di matchmaking](#)
- Integrare FlexMatch nel tuo servizio client di gioco. Aggiungi funzionalità al servizio client di gioco per avviare nuove sessioni di gioco con abbinamento. Le richieste di abbinamento specificano quale matchmaker utilizzare e forniscono i dati del giocatore necessari per l'abbinamento. Vedi questo argomento:
 - [Add \(Aggiungi\) FlexMatch a un client di gioco](#)
- Integrare FlexMatch nel tuo server di gioco. Aggiungi funzionalità al tuo server di gioco per avviare le sessioni di gioco create tramite l'abbinamento. Le richieste per questo tipo di sessione di gioco includono informazioni specifiche sull'abbinamento, inclusi giocatori e assegnazioni dei team. Il server di gioco deve accedere e utilizzare queste informazioni durante la costruzione di una sessione di gioco per l'abbinamento. Vedi questo argomento:
 - [Add \(Aggiungi\) FlexMatch a un Amazon GameLift Servers-server di gioco ospitato](#)
- Configurare FlexMatch riempimento (opzionale). Richiedi ulteriori abbinamenti di giocatori per riempire gli slot giocatori aperti nei giochi esistenti. Puoi attivare il riempimento automatico per avere Amazon GameLift Servers gestire le richieste di riempimento. In alternativa puoi gestire il backfill manualmente aggiungendo la funzionalità al servizio client di gioco o al server di gioco per avviare richieste di backfill degli abbinamenti. Vedi questo argomento:
 - [Riempi i giochi esistenti con FlexMatch](#)

Note

FlexMatch il backfill non è attualmente disponibile per i giochi che utilizzano Amazon GameLift Servers In tempo reale.

Costruire un Amazon GameLift Servers FlexMatch parainfo

Questa sezione descrive gli elementi chiave di un matchmaker e come crearne e personalizzarne uno per il tuo gioco. Ciò include la configurazione di una configurazione di matchmaking e di un set di regole di matchmaking.

Creare il tuo matchmaker è il primo passo del FlexMatch tabelle di marcia:

- [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#)
- [Tabella di marcia: Crea una soluzione di matchmaking autonoma con FlexMatch](#)

A FlexMatch matchmaker si occupa di costruire una partita di gioco. Gestisce il pool di richieste di matchmaking ricevute, elabora e seleziona i giocatori per trovare i migliori gruppi di giocatori possibili, e forma le squadre per una partita. Per giochi che utilizzano Amazon GameLift Servers per ospitare, posiziona e avvia anche una sessione di gioco per la partita.

FlexMatch abbina il servizio di matchmaking a un motore di regole personalizzabile. Ciò ti consente di progettare come abbinare i giocatori in base agli attributi dei giocatori e alle modalità di gioco più adatte al tuo gioco, e fai affidamento su FlexMatch per gestire le fasi principali della formazione di gruppi di giocatori e del loro inserimento nelle partite. Vedi maggiori dettagli sul matchmaking personalizzato in [FlexMatch esempi di set di regole](#).

Dopo aver formato una partita, FlexMatch fornisce i dati della partita per il posizionamento della sessione di gioco. Per i giochi che utilizzano Amazon GameLift Servers per ospitare, FlexMatch invia una richiesta di posizionamento di una sessione di gioco con giocatori abbinati alla coda di una sessione di gioco. La coda cerca le risorse di hosting disponibili sul tuo Amazon GameLift Servers fugge e inizia una nuova sessione di gioco per la partita. Per i giochi che utilizzano un'altra soluzione di hosting, FlexMatch fornisce i dati delle partite da inserire nel componente di posizionamento della sessione di gioco.

Per una descrizione dettagliata di come FlexMatch matchmaker elabora le richieste di matchmaking che riceve, vedere. [FlexMatch processo di matchmaking](#)

Argomenti

- [Progetta un FlexMatch parainfo](#)
- [Costruisci un FlexMatch set di regole](#)

- [Crea una configurazione di matchmaking](#)
- [Configurazione FlexMatch notifiche di eventi](#)

Progetta un FlexMatch parainfo

Questo argomento fornisce indicazioni su come progettare un matchmaker adatto al tuo gioco.

Argomenti

- [Configura un matchmaker di base](#)
- [Scegli una posizione per il matchmaker](#)
- [Aggiungi elementi opzionali](#)

Configura un matchmaker di base

Come minimo, un matchmaker necessita dei seguenti elementi:

- Il set di regole determina la dimensione e la portata dei team per una partita e definisce un set di regole da utilizzare quando si valutano i giocatori per una partita. Ogni matchmaker è configurato per l'utilizzo di un solo set di regole. Consulta [Costruisci un FlexMatch set di regole](#) e [FlexMatch esempi di set di regole](#).
- Il destinatario della notifica riceve tutte le notifiche degli eventi di matchmaking. Devi configurare un argomento Amazon Simple Notification Service (SNS) e quindi aggiungere l'ID dell'argomento al matchmaker. Per ulteriori informazioni su come impostare le notifiche, consulta [Configurazione FlexMatch notifiche di eventi](#).
- Il timeout della richiesta determina la durata con cui le richieste di abbinamento possono rimanere nel pool di richiesta ed essere valutate per potenziali partite. Se una richiesta scade, significa che non è riuscita a creare una partita e viene rimossa dal pool.
- Quando si utilizza FlexMatch con Amazon GameLift Servers hosting gestito, la coda delle sessioni di gioco trova le migliori risorse disponibili per ospitare una sessione di gioco per la partita e avvia una nuova sessione di gioco. Ogni coda è configurata con un elenco di posizioni e tipi di risorse (comprese le istanze Spot o On-Demand) che determinano dove è possibile effettuare le sessioni di gioco. [Per ulteriori informazioni sulle code, consulta Utilizzo delle code con più sedi](#).

Scegli una posizione per il matchmaker

Decidi dove desideri che si svolga l'attività di matchmaking e crea la configurazione di matchmaking e il set di regole in quella località. Amazon GameLift Servers gestisce i pool di ticket per le richieste di partite del tuo gioco, dove vengono ordinate e valutate per verificare se ci sono partite valide. Dopo aver creato una partita, Amazon GameLift Servers invia i dettagli della partita per il posizionamento della sessione di gioco. Puoi eseguire le sessioni di gioco corrispondenti in qualsiasi luogo supportato dalla tua soluzione di hosting.

Scopri [FlexMatch supportata Regioni AWS](#) i luoghi in cui puoi creare FlexMatch risorse.

Quando ne scegli una Regione AWS per il tuo matchmaker, considera in che modo la posizione potrebbe influire sulle prestazioni e come può ottimizzare l'esperienza di partita per i giocatori. È preferibile seguire le best practice seguenti:

- Posiziona un matchmaker in un luogo vicino ai tuoi giocatori e al tuo servizio clienti che invia FlexMatch richieste di matchmaking. Questo approccio riduce l'effetto di latenza sul flusso di lavoro delle richieste di matchmaking e lo rende più efficiente.
- Se il tuo gioco raggiunge un pubblico globale, prendi in considerazione la possibilità di creare matchmaker in più località e di inoltrare le richieste di partita al matchmaker più vicino al giocatore. Oltre a incrementare l'efficienza, ciò fa sì che si formino pool di ticket con giocatori geograficamente vicini tra loro, il che migliora la capacità del matchmaker di abbinare i giocatori in base ai requisiti di latenza.
- Quando si utilizza FlexMatch con Amazon GameLift Servers hosting gestito, posiziona il tuo matchmaker e la coda della sessione di gioco che utilizza nella stessa posizione. Questo aiuta a ridurre al minimo la latenza di comunicazione tra matchmaker e coda.

Aggiungi elementi opzionali

Oltre a questi requisiti minimi, è possibile configurare il matchmaker con le seguenti opzioni aggiuntive. Se stai usando FlexMatch con un Amazon GameLift Servers soluzione di hosting, molte funzionalità sono integrate. Se stai usando FlexMatch come servizio di matchmaking autonomo, potresti voler integrare queste funzionalità nel tuo sistema.

Accettazione da parte dei giocatori

Puoi configurare un matchmaker in modo che richieda che tutti i giocatori selezionati per una partita debbano accettare la partecipazione. Se il sistema richiede l'accettazione, a tutti i giocatori deve

essere data la possibilità di accettare o rifiutare una partita proposta. Una partita deve ottenere l'accettazione da tutti i giocatori nella partita proposta prima di poter essere completata. Se un giocatore rifiuta o non riesce ad accettare una partita, la partita proposta viene scartata e i ticket vengono gestiti come segue. I biglietti per i quali tutti i giocatori presenti nel ticket hanno accettato la partita vengono restituiti al matchmaking pool per essere processati ulteriormente. I ticket in cui almeno un giocatore ha rifiutato la partita o non ha risposto vengono messi in stato di errore e non vengono più elaborati. L'accettazione da parte del giocatore richiede un limite di tempo; tutti i giocatori devono accettare una partita proposta entro il limite di tempo affinché la partita possa continuare.

Modalità di backfill

Utilizzo FlexMatch riempi le sessioni di gioco per mantenere le sessioni di gioco piene di nuovi giocatori ben preparati per tutta la durata della sessione di gioco. Quando si gestiscono le richieste di riempimento, FlexMatch utilizza lo stesso matchmaker usato per abbinare i giocatori originali. Puoi personalizzare il modo in cui viene data priorità ai ticket di riserva con i biglietti per le nuove partite, posizionando i biglietti di riserva in prima fila o in fondo alla fila. Ciò significa che, man mano che nuovi giocatori entrano nel pool di matchmaking, è più o meno probabile che vengano inseriti in una partita esistente piuttosto che in una partita appena creata.

Il riempimento manuale è disponibile indipendentemente dal fatto che il gioco lo utilizzi FlexMatch con gestito Amazon GameLift Servers hosting o con altre soluzioni di hosting. Il backfill manuale offre la flessibilità di decidere quando attivare una richiesta di backfill. Ad esempio, potresti voler aggiungere nuovi giocatori solo durante determinate fasi del gioco o solo quando esistono determinate condizioni.

Il riempimento automatico è disponibile solo per i giochi che utilizzano la modalità gestita Amazon GameLift Servers hosting. Con questa funzionalità abilitata, se una sessione di gioco inizia con slot per giocatori aperti, Amazon GameLift Servers inizia a generare automaticamente le relative richieste di riempimento. Questa funzione ti consente di impostare il matchmaking in modo che le nuove partite vengano avviate con un numero minimo di giocatori e poi completate rapidamente man mano che nuovi giocatori entrano nel pool di matchmaking. Puoi disattivare il riempimento automatico in qualsiasi momento durante la durata della sessione di gioco.

Proprietà del gioco

Per giochi che utilizzano FlexMatch con Amazon GameLift Servers hosting gestito, puoi fornire informazioni aggiuntive da passare a un server di gioco ogni volta che viene richiesta una nuova sessione di gioco. Questo può essere un modo utile per passare alle configurazioni delle modalità di gioco necessarie per avviare una sessione di gioco in base al tipo di partite che si sta creando. Tutte

le sessioni di gioco per le partite create da un matchmaker ricevono lo stesso set di proprietà di gioco. Puoi variare le informazioni sulle proprietà del gioco creando diverse configurazioni di matchmaking.

Slot giocatori riservati

È possibile stabilire che determinate slot giocatori in ogni partita vengano riservati e compilati in un secondo momento. Questo è possibile configurando la proprietà di "conteggio giocatore aggiuntivo" di una configurazione matchmaking.

Personalizzazione di dati eventi

Utilizzare questa proprietà per includere una serie di informazioni personalizzate in tutti gli eventi correlati al matchmaker. Questa funzione può essere utile per monitorare alcune attività univoche per il gioco, tra cui monitorare le prestazioni dei matchmaker.

Costruisci un FlexMatch set di regole

Ogni FlexMatch il matchmaker deve avere un set di regole. Il set di regole determina i due elementi chiave di un abbinamento: la struttura e le dimensioni dei team del tuo gioco e la modalità di raggruppamento dei giocatori per il migliore abbinamento possibile.

Ad esempio, un set di regole potrebbe descrivere una corrispondenza come questa: crea una corrispondenza con due squadre di cinque giocatori ciascuna, una costituita da difensori e un'altra da invasori. Una squadra può avere giocatori alle prime armi ed esperti, ma l'abilità media delle due squadre deve essere entro 10 punti l'una dall'altra. Se non viene effettuato alcun abbinamento dopo 30 secondi, rendi gradualmente più flessibili i requisiti di abilità.

Gli argomenti in questa sezione descrivono come progettare e creare un set di regole di abbinamento. Quando si crea un set di regole, è possibile utilizzare uno dei Amazon GameLift Servers console o AWS CLI.

Argomenti

- [Progetta un FlexMatch set di regole](#)
- [Progetta un FlexMatch set di regole per partite di grandi dimensioni](#)
- [Tutorial: Crea un set di regole di matchmaking](#)
- [FlexMatch esempi di set di regole](#)

Progetta un FlexMatch set di regole

Questo argomento tratta la struttura di base di un set di regole e come creare un set di regole per piccole partite fino a 40 giocatori. Un set di regole di matchmaking fa due cose: definisce la struttura e le dimensioni della squadra di una partita e indica al matchmaker come scegliere i giocatori per formare la migliore partita possibile.

Ma il tuo set di regole di matchmaking può fare di più. Ad esempio, puoi:

- Ottimizzi l'algoritmo di matchmaking per il tuo gioco.
- Imposti i requisiti minimi di latenza dei giocatori per proteggere la qualità del gioco.
- Riduci gradualmente i requisiti di squadra e le regole delle partite nel tempo, in modo che tutti i giocatori attivi possano trovare una partita accettabile quando ne vogliono una.
- Definisci la gestione delle richieste di matchmaking di gruppo utilizzando l'aggregazione dei gruppi.
- Elabori partite di grandi dimensioni con 40 o più giocatori. Per ulteriori informazioni sulla creazione di partite di grandi dimensioni, vedi [Progetta un FlexMatch set di regole per partite di grandi dimensioni](#).

Quando crei un set di regole di matchmaking, considera le seguenti attività facoltative e obbligatorie:

- [Descrivi il set di regole \(obbligatorio\)](#)
- [Personalizza l'algoritmo di abbinamento](#)
- [Dichiara gli attributi del giocatore](#)
- [Definisci le squadre di gioco](#)
- [Stabilisci le regole per l'abbinamento dei giocatori](#)
- [Consenti ai requisiti di attenuarsi nel tempo](#)

Puoi creare il tuo set di regole usando il Amazon GameLift Servers console o l'[CreateMatchmakingRuleSet](#) operazione.

Descrivi il set di regole (obbligatorio)

Fornisci i dettagli per il set di regole.

- `nome` (opzionale) — Un'etichetta descrittiva per uso personale. Questo valore non è associato al nome del set di regole specificato durante la creazione del set di regole con Amazon GameLift Servers.
- `ruleLanguageVersion`— La versione del linguaggio delle espressioni di proprietà utilizzato per creare FlexMatch regole. Il valore deve essere `1.0`.

Personalizza l'algoritmo di abbinamento

FlexMatch ottimizza l'algoritmo predefinito per la maggior parte dei giochi per far sì che i giocatori accedano a partite accettabili con tempi di attesa minimi. Puoi personalizzare l'algoritmo e adattare il matchmaking al tuo gioco.

Quanto segue è l'impostazione predefinita FlexMatch algoritmo di matchmaking:

1. FlexMatch inserisce tutti i ticket di matchmaking aperti e i ticket di riserva in un pool di biglietti.
2. FlexMatch raggruppa casualmente i biglietti del pool in uno o più lotti. Man mano che il pool di biglietti diventa più grande, FlexMatch forma lotti aggiuntivi per mantenere una dimensione ottimale del lotto.
3. FlexMatch ordina i biglietti per età, all'interno di ciascun lotto.
4. FlexMatch crea una partita sulla base del ticket più vecchio di ogni lotto.

Per personalizzare l'algoritmo di corrispondenza, aggiungi un `algorithm` componente allo schema del set di regole. Vedi [FlexMatch schema del set di regole](#) per le informazioni di riferimento complete.

Utilizza le seguenti personalizzazioni opzionali per influire sulle diverse fasi del processo di matchmaking.

- [Aggiungi l'ordinamento pre-batch](#)
- [Batch di moduli basati sugli attributi `BatchDistance`](#)
- [Dai priorità ai ticket di riempimento](#)
- [Preferisci i biglietti più vecchi con le espansioni](#)

Aggiungi l'ordinamento pre-batch

Puoi ordinare il pool di biglietti prima di formare i batch. Questo tipo di personalizzazione è più efficace con giochi con grandi pool di biglietti. L'ordinamento pre-batch può aiutare ad accelerare il processo di matchmaking e ad aumentare l'uniformità dei giocatori nelle caratteristiche definite.

Definisci i metodi di ordinamento pre-batch utilizzando la proprietà `algorithm.batchingPreference`. L'impostazione predefinita è `random`.

Le opzioni per personalizzare l'ordinamento pre-batch includono:

- Ordina per attributi del giocatore. Fornisci un elenco di attributi del giocatore per preordinare il pool di biglietti.

Per ordinare in base agli attributi del giocatore, imposta `batchingPreference` e definisci l'elenco degli attributi del giocatore in `sortByAttributes.sorted`. Per usare un attributo, devi prima dichiararlo nel `playerAttributes` componente del set di regole.

Nell'esempio seguente, FlexMatch ordina il pool di biglietti in base alla mappa di gioco preferita dai giocatori e quindi in base alle abilità del giocatore. È più probabile che i batch risultanti contengano giocatori con abilità simili che desiderano utilizzare la stessa mappa.

```
"algorithm": {
  "batchingPreference": "sorted",
  "sortByAttributes": ["map", "player_skill"],
  "strategy": "exhaustiveSearch"
},
```

- Ordina per latenza. Crea corrispondenze con la latenza più bassa disponibile o crea rapidamente corrispondenze con una latenza accettabile. Questa personalizzazione è utile per i set di regole che formano partite di grandi dimensioni con più di 40 giocatori.

Imposta la proprietà dell'algoritmo `strategy.subbalanced`. La strategia bilanciata limita i tipi disponibili di istruzioni delle regole. Per ulteriori informazioni, consulta [Progetta un FlexMatch set di regole per partite di grandi dimensioni](#).

FlexMatch ordina i ticket in base ai dati di latenza riportati dai giocatori in uno dei seguenti modi:

- posizioni con la latenza più bassa. Il pool di biglietti è preordinato in base alle località in cui i giocatori segnalano i valori di latenza più bassi. FlexMatch quindi raggruppa i ticket a bassa latenza nelle stesse posizioni, creando un'esperienza di gioco migliore. Riduce anche il numero

di ticket in ogni lotto, quindi il matchmaking può richiedere più tempo. Per utilizzare questa personalizzazione, imposta `batchingPreference` su `fastestRegion`, come mostrato nell'esempio seguente.

```
"algorithm": {
  "batchingPreference": "fastestRegion",
  "strategy": "balanced"
},
```

- La latenza accettabile corrisponde rapidamente. Il pool di biglietti è preordinato in base alle località in cui i giocatori segnalano un valore di latenza accettabile. Ciò forma un minor numero di lotti contenenti più biglietti. Con più biglietti in ogni lotto, trovare partite accettabili è più veloce. Per utilizzare questa personalizzazione, impostate la proprietà `batchingPreference` su `largestPopulation`, come illustrato nell'esempio seguente.

```
"algorithm": {
  "batchingPreference": "largestPopulation",
  "strategy": "balanced"
},
```

Note

Il valore predefinito per la strategia bilanciata è `largestPopulation`.

Dai priorità ai ticket di riempimento

Se il tuo gioco implementa il riempimento automatico o il riempimento manuale, puoi personalizzare come FlexMatch elabora i ticket di matchmaking in base al tipo di richiesta. Il tipo di richiesta può essere una nuova richiesta di abbinamento o di riempimento. Per impostazione predefinita, FlexMatch tratta entrambi i tipi di richieste allo stesso modo.

La prioritizzazione del backfill influisce su come FlexMatch gestisce i ticket dopo averli raggruppati. La prioritizzazione dei backfill richiede set di regole per utilizzare la strategia di ricerca esaustiva.

FlexMatch non abbinna più ticket di backfill insieme.

Per modificare la priorità per i ticket di riempimento, imposta la proprietà `backfillPriority`

- Abbinare prima i biglietti di backfill. Questa opzione cerca di abbinare i ticket di riserva prima di creare nuove partite. Ciò significa che i giocatori entranti hanno maggiori possibilità di partecipare a una partita esistente.

È preferibile utilizzarlo se il gioco utilizza il backfill automatico. Il riempimento automatico viene spesso utilizzato nei giochi con sessioni di gioco brevi e tempi di risposta elevati. Il backfill automatico aiuta questi giochi a creare un numero minimo di partite possibili e a farle iniziare mentre FlexMatch cerca altri giocatori per riempire gli slot aperti.

Imposta `backfillPriority` su `high`.

```
"algorithm": {
  "backfillPriority": "high",
  "strategy": "exhaustiveSearch"
},
```

- I biglietti Match Backfill sono esauriti. Questa opzione ignora i ticket di backfill finché non valuta tutti gli altri ticket. Ciò significa che FlexMatch inserisce i giocatori entranti nelle partite esistenti quando non riesce ad abbinarli a nuove partite.

Questa opzione è utile quando desideri utilizzare il backfill come ultima possibilità per coinvolgere i giocatori in una partita, ad esempio quando non ci sono abbastanza giocatori per formare una nuova partita.

Imposta `backfillPriority` su `low`.

```
"algorithm": {
  "backfillPriority": "low",
  "strategy": "exhaustiveSearch"
},
```

Preferisci i biglietti più vecchi con le espansioni

Le regole di espansione allentano i criteri di abbinamento quando le partite sono difficili da completare. Amazon GameLift Servers applica le regole di espansione quando i biglietti di una partita parzialmente completata raggiungono una certa età. I timestamp di creazione dei biglietti determinano quando Amazon GameLift Servers applica le regole; per impostazione predefinita, FlexMatch tiene traccia del timestamp del ticket abbinato più di recente.

Per cambiare quando FlexMatch applica le regole di espansione, imposta la proprietà `expansionAgeSelection` come segue:

- Espandi in base ai biglietti più recenti. Questa opzione applica le regole di espansione in base al ticket più recente aggiunto alla potenziale partita. Ogni volta FlexMatch corrisponde a un nuovo ticket, l'orologio viene azzerato. Con questa opzione, le corrispondenze risultanti tendono ad essere di qualità superiore, ma richiedono più tempo per essere abbinati; le richieste di abbinamento potrebbero scadere prima del completamento se impiegano troppo tempo a corrispondere. Impostato `expansionAgeSelection` su `sunewest`. `newest` è l'impostazione predefinita.
- Espandi in base ai ticket più vecchi. Questa opzione applica regole di espansione basate sul ticket più vecchio della potenziale partita. Con questa opzione, FlexMatch applica le espansioni più velocemente, il che migliora i tempi di attesa per i primi giocatori abbinati, ma riduce la qualità della partita per tutti i giocatori. Imposta `expansionAgeSelection` su `oldest`.

```
"algorithm": {  
  "expansionAgeSelection": "oldest",  
  "strategy": "exhaustiveSearch"  
},
```

Dichiara gli attributi del giocatore

In questa sezione, elenca gli attributi dei singoli giocatori da includere nelle richieste di matchmaking. Esistono due motivi per cui potresti dichiarare gli attributi dei giocatori in un set di regole:

- Quando il set di regole contiene regole che si basano sugli attributi del giocatore.
- Quando vuoi passare un attributo del giocatore alla sessione di gioco tramite la richiesta della partita. Ad esempio, potresti voler passare le scelte dei personaggi dei giocatori alla sessione di gioco prima che ogni giocatore si connetta.

Nella dichiarazione di un attributo di un giocatore, includere le informazioni riportate di seguito:

- `name` (obbligatorio): questo valore deve essere univoco per il set di regole.
- `type` (obbligatorio) — Il tipo di dati del valore dell'attributo. I tipi di dati validi sono `numero`, `stringa`, `elenco di stringhe` o `mappa di stringhe`.

- **default** (opzionale): immetti un valore predefinito da utilizzare se una richiesta di matchmaking non fornisce un valore di attributo. Se non viene dichiarato alcun valore predefinito e una richiesta non include un valore, FlexMatch non può soddisfare la richiesta.

Definisci le squadre di gioco

Descrivi la struttura e le dimensioni dei team per un abbinamento. Ogni abbinamento deve avere almeno un team e puoi definire tutti i team che desideri. I tuoi team possono avere lo stesso numero di giocatori o essere asimmetrici. Ad esempio, potresti definire un team di mostri con un giocatore singolo e un team di cacciatori con 10 giocatori.

FlexMatch elabora le richieste di partite come partite piccole o grandi, in base a come il set di regole definisce le dimensioni delle squadre. Le partite potenziali con un massimo di 40 giocatori sono partite di breve durata, quelle con più di 40 giocatori sono partite importanti. Per determinare la dimensione dell'abbinamento potenziale di un set di regole, aggiungi le impostazioni `maxPlayer` per tutti i team definiti nel set di regole.

- **nome** (obbligatorio): assegna a ogni squadra un nome univoco. Utilizzi questo nome nelle regole e nelle espansioni e FlexMatch riferimenti per i dati di matchmaking in una sessione di gioco.
- **MaxPlayers** (obbligatorio): specifica il numero massimo di giocatori da assegnare alla squadra.
- **MinPlayers** (obbligatorio): specifica il numero minimo di giocatori da assegnare alla squadra.
- **quantità** (opzionale): specifica il numero di squadre da formare con questa definizione. Quando FlexMatch crea una partita, dà a queste squadre il nome fornito con un numero aggiunto. Ad esempio `Red-Team1Red-Team2, eRed-Team3`.

FlexMatch tenta di riempire le squadre fino al numero massimo di giocatori, ma crea squadre con meno giocatori. Se vuoi che tutti i team dell'abbinamento abbiano le stesse dimensioni, puoi creare una regola apposita. Vedi l'[FlexMatch esempi di set di regole](#) argomento per un esempio di `EqualTeamSizes` regola.

Stabilisci le regole per l'abbinamento dei giocatori

Crea una serie di regole che valutino l'accettazione dei giocatori in una partita. Le regole possono stabilire i requisiti che si applicano a singoli giocatori, team o un intero abbinamento. Quando Amazon GameLift Servers elabora una richiesta di partita, inizia con il giocatore più anziano del gruppo di giocatori disponibili e costruisce una partita attorno a quel giocatore. Per informazioni dettagliate sulla creazione FlexMatch regole, vedi [FlexMatch tipi di regole](#).

- **name** (obbligatorio) — Un nome significativo che identifica in modo univoco la regola all'interno di un set di regole. I nomi delle regole sono anche presenti nei log di eventi e parametri in grado di monitorare le attività correlate a questa regola.
- **descrizione** (opzionale) — Utilizzate questo elemento per allegare una descrizione testuale in formato libero.
- **type** (obbligatorio) — L'elemento `type` identifica l'operazione da utilizzare durante l'elaborazione della regola. Ogni tipo di regola richiede un set di proprietà aggiuntive. Consulta l'elenco delle proprietà e dei tipi di regole validi disponibile in [FlexMatch linguaggio delle regole](#).
- **Proprietà del tipo di regola** (potrebbe essere obbligatoria): a seconda del tipo di regola definita, potrebbe essere necessario impostare determinate proprietà della regola. Scopri di più sulle proprietà e su come utilizzare il FlexMatch linguaggio di espressione delle proprietà in [FlexMatch linguaggio delle regole](#).

Consenti ai requisiti di attenuarsi nel tempo

Le espansioni consentono di allentare i criteri delle regole nel tempo quando FlexMatch non riesce a trovare una corrispondenza. Questa funzionalità garantisce che FlexMatch rende disponibile il meglio quando non riesce a creare un abbinamento perfetto. Allentando le regole con un'espansione, si amplia gradualmente il numero di giocatori che rappresentano una partita accettabile.

Le espansioni iniziano quando l'età del biglietto più recente nella partita incompleta corrisponde al tempo di attesa per l'espansione. Quando FlexMatch aggiunge un nuovo ticket alla partita, l'orologio del tempo di attesa per l'espansione potrebbe essere azzerato. Puoi personalizzare l'inizio delle espansioni nella `algorithm` sezione del set di regole.

Ecco un esempio di espansione che aumenta gradualmente il livello minimo di abilità richiesto per la partita. Il set di regole utilizza una regola sulla distanza, denominata in `SkillDelta` modo da richiedere che tutti i giocatori di una partita si trovino a meno di 5 livelli di abilità l'uno dall'altro. Se non vengono create nuove partite per quindici secondi, questa espansione rileva una differenza di livello di abilità di 10, quindi dieci secondi dopo una differenza di 20.

```
"expansions": [{
  "target": "rules[SkillDelta].maxDistance",
  "steps": [{
    "waitTimeSeconds": 15,
    "value": 10
  }, {
    "waitTimeSeconds": 25,
```

```
        "value": 20
    }]
}]
```

Con i matchmaker che hanno abilitato il riempimento automatico, non limitarti troppo rapidamente ai requisiti di conteggio dei giocatori. Sono necessari alcuni secondi per l'avvio della nuova sessione di gioco e l'avvio del backfill. Un approccio migliore consiste nell'avviare l'espansione dopo che il riempimento automatico tende a entrare in gioco nei giochi. I tempi di espansione variano a seconda della composizione della squadra, quindi fai dei test per trovare la strategia di espansione migliore per il tuo gioco.

Progetta un FlexMatch set di regole per partite di grandi dimensioni

Se il tuo set di regole crea partite che ammettono da 41 a 200 giocatori, devi apportare alcune modifiche alla configurazione del set di regole. Queste modifiche ottimizzano l'algoritmo delle partite in modo che possa creare partite di grandi dimensioni e allo stesso tempo ridurre i tempi di attesa dei giocatori. Di conseguenza, set di regole per partite di grandi dimensioni sostituiscono le lunghe regole personalizzate con soluzioni standard ottimizzate per le priorità di matchmaking comuni.

Ecco come determinare se è necessario ottimizzare il set di regole per partite di grandi dimensioni:

1. Per ogni squadra definita nel tuo set di regole, ottieni il valore di MaxPlayer,
2. Somma tutti i valori di MaxPlayer. Se il totale supera i 40, hai un ampio set di regole per le partite.

Per ottimizzare il set di regole per partite di grandi dimensioni, apporta le modifiche descritte di seguito. Vedi lo schema per un set di regole per partite di grandi dimensioni [Schema del set di regole per partite di grandi dimensioni](#) ed esempi di set di regole in [Esempio: crea una partita di grandi dimensioni](#).

Personalizza l'algoritmo di corrispondenza per corrispondenze di grandi dimensioni

Aggiungi un componente dell'algoritmo al set di regole, se non ne esiste già uno. Imposta le seguenti proprietà.

- `strategy`(obbligatorio) — Imposta la `strategy` proprietà su «bilanciata». Questa impostazione si attiva FlexMatch per effettuare ulteriori controlli post-partita per trovare l'equilibrio ottimale della squadra in base a uno specifico attributo del giocatore, definito nella `balancedAttribute` proprietà. La strategia bilanciata sostituisce la necessità di regole personalizzate per creare squadre equamente abbinata.

- `balancedAttribute`(obbligatorio): identifica un attributo del giocatore da utilizzare per bilanciare le squadre in una partita. Questo attributo deve avere un tipo di dati numerico (doppio o intero). Ad esempio, se scegli di bilanciare l'abilità del giocatore, FlexMatch cerca di assegnare i giocatori in modo che tutte le squadre abbiano livelli di abilità aggregati il più equi possibile. L'attributo di bilanciamento deve essere dichiarato negli attributi del giocatore del set di regole.
- `batchingPreference`(opzionale): scegli quanta enfasi vuoi dare alla creazione di partite con la latenza più bassa possibile per i tuoi giocatori. Questa impostazione influisce sul modo in cui i biglietti delle partite vengono ordinati prima di creare le partite. Le opzioni includono:
 - Popolazione più numerosa. FlexMatch consente le partite utilizzando tutti i ticket del pool che hanno valori di latenza accettabili in almeno una località in comune. Di conseguenza, il potenziale pool di ticket tende ad essere ampio, il che rende più facile riempire le partite più rapidamente. I giocatori potrebbero essere inseriti in partite con una latenza accettabile, ma non sempre ottimale. Se la `batchingPreference` proprietà non è impostata, questo è il comportamento predefinito quando `strategy` è impostata su «bilanciato».
 - Posizione più veloce. FlexMatch preordina tutti i ticket del pool in base a dove riportano i valori di latenza più bassi. Di conseguenza, le partite tendono a essere organizzate con giocatori che segnalano una bassa latenza nelle stesse località. Allo stesso tempo, il potenziale pool di biglietti per ogni partita è inferiore, il che può aumentare il tempo necessario per riempire una partita. Inoltre, poiché viene data maggiore priorità alla latenza, i giocatori nelle partite possono variare maggiormente per quanto riguarda l'attributo di bilanciamento.

L'esempio seguente configura l'algoritmo della partita in modo che si comporti come segue: (1) Preordina il pool di ticket per raggruppare i ticket in base alla località in cui hanno valori di latenza accettabili; (2) Forma lotti di ticket ordinati per l'abbinamento; (3) Crea partite con biglietti in batch e bilancia le squadre per uniformare l'abilità media del giocatore.

```
"algorithm": {
  "strategy": "balanced",
  "balancedAttribute": "player_skill",
  "batchingPreference": "largestPopulation"
},
```

Dichiara gli attributi del giocatore

Assicurati di dichiarare l'attributo `player` utilizzato come attributo di bilanciamento nell'algoritmo del set di regole. Questo attributo deve essere incluso per ogni giocatore in una richiesta di matchmaking.

Puoi fornire un valore predefinito per l'attributo `player`, ma il bilanciamento degli attributi funziona meglio quando vengono forniti valori specifici per il giocatore.

Definisci le squadre

Il processo di definizione delle dimensioni e della struttura della squadra è lo stesso delle partite di piccole dimensioni, ma è così FlexMatch il numero di membri delle squadre è diverso. Ciò influisce sulla probabilità che le partite siano riempite solo parzialmente. In risposta, potresti voler modificare le dimensioni minime della tua squadra.

FlexMatch utilizza le seguenti regole quando assegna un giocatore a una squadra. In primo luogo: cerca i team che non hanno ancora raggiunto il loro requisito minimo di giocatori. In secondo luogo: di quei team, individua quello con il maggior numero di slot aperti.

Per gli abbinamenti che definiscono più team di uguali dimensioni, i giocatori vengono aggiunti in sequenza a ciascun team fino al completamento. Di conseguenza, le squadre in una partita hanno sempre un numero di giocatori quasi uguale, anche quando la partita non è completa. Non è attualmente possibile forzare team di uguali dimensioni in abbinamenti grandi. Per gli abbinamenti con team di dimensioni asimmetriche, il processo è un po' più complesso. In questo scenario, i giocatori vengono inizialmente assegnati alle squadre più grandi con il maggior numero di slot disponibili. Man mano che il numero di slot aperti diventa più equamente distribuito tra tutte le squadre, i giocatori vengono inseriti nelle squadre più piccole.

Ad esempio, supponiamo che tu abbia un set di regole con tre squadre. Le squadre rosse e blu sono entrambe impostate su `maxPlayers = 10`, `minPlayers = 5`. La squadra verde è impostata su `maxPlayers = 3`, `minPlayers = 2`. Ecco la sequenza di riempimento:

1. Nessuna squadra è arrivata a `minPlayers`. I team rosso e blu hanno 10 slot aperti, mentre il verde ne ha 3. I primi 10 giocatori vengono assegnati (5 ciascuno) ai team rosso e blu. Entrambe le squadre sono ora arrivate a `minPlayers`.
2. La squadra verde non è ancora arrivata a `minPlayers`. I successivi 2 giocatori vengono assegnati al team verde. La squadra dei Verdi è ora arrivata a `minPlayers`.
3. Con tutte le squadre presentate a `minPlayers`, ora vengono assegnati giocatori aggiuntivi in base al numero di slot disponibili. Le squadre rosse e blu hanno ciascuna 5 slot aperte, mentre la squadra verde ne ha 1. I successivi 8 giocatori vengono assegnati (4 ciascuno) alle squadre rosse e blu. Tutte le squadre ora hanno 1 posto libero.
4. I restanti 3 posti per giocatori vengono assegnati (1 ciascuno) alle squadre senza un ordine particolare.

Stabilisci delle regole per partite di grandi dimensioni

Il matchmaking per partite di grandi dimensioni si basa principalmente sulla strategia di bilanciamento e sulle ottimizzazioni del batch di latenza. La maggior parte delle regole personalizzate non sono disponibili. Tuttavia, puoi incorporare i seguenti tipi di regole:

- Regola che stabilisce un limite rigido alla latenza dei giocatori. Usa il tipo di `Latency` regola con la proprietà `maxLatency`. Vedi [Latency rule \(Regola di latenza\)](#) riferimento. Ecco un esempio che imposta la latenza giocatore massima su 200 millisecondi:

```
"rules": [{
  "name": "player-latency",
  "type": "latency",
  "maxLatency": 200
}],
```

- Regola per raggruppare i giocatori in base alla vicinanza in uno specifico attributo del giocatore. Questo è diverso dalla definizione di un attributo di bilanciamento come parte dell'algoritmo per partite di grandi dimensioni, che si concentra sulla creazione di squadre con partite uguali. Questa regola raggruppa i ticket di matchmaking in base alla somiglianza dei valori degli attributi specificati, come l'abilità principiante o esperto, il che tende a far sì che i giocatori delle partite siano strettamente allineati sull'attributo specificato. Usa il tipo di `batchDistance` regola, identifica un attributo basato su numeri e specifica l'intervallo più ampio da consentire. Vedi riferimento. [Regola della distanza in batch](#) Ecco un esempio che richiede che i giocatori di una partita si trovino all'interno di un livello di abilità l'uno dall'altro:

```
"rules": [{
  "name": "batch-skill",
  "type": "batchDistance",
  "batchAttribute": "skill",
  "maxDistance": 1
}],
```

Allenta i requisiti delle partite di grandi dimensioni

Come per gli abbinamenti piccoli, puoi utilizzare le espansioni per rendere più flessibili i requisiti degli abbinamenti nel tempo, quando non è possibile effettuare abbinamenti validi. Nelle partite di grandi dimensioni, hai la possibilità di allentare le regole di latenza o il numero di giocatori di squadra.

Se utilizzi il riempimento automatico delle partite per partite di grandi dimensioni, evita di abbassare troppo rapidamente il conteggio dei giocatori della tua squadra. FlexMatch inizia a generare richieste di riempimento solo dopo l'inizio di una sessione di gioco, cosa che potrebbe non avvenire per diversi secondi dopo la creazione di una partita. Durante quel periodo, FlexMatch può creare più nuove sessioni di gioco parzialmente riempite, specialmente quando le regole del conteggio dei giocatori vengono abbassate. Di conseguenza, ti ritrovi con più sessioni di gioco di quelle che ti servono e i giocatori si diffondono troppo raramente. Si consiglia di concedere al primo passaggio dell'espansione del numero dei giocatori un tempo di attesa più lungo, abbastanza lungo da consentire l'avvio della sessione di gioco. Dal momento che le richieste di backfill hanno la priorità più alta con gli abbinamenti grandi, i giocatori in arrivo verranno inseriti nei giochi esistenti prima di iniziare un nuovo gioco. Potrebbe essere necessario effettuare delle prove per trovare il tempo di attesa ideale per il tuo gioco.

Ecco un esempio che riduce gradualmente il numero dei giocatori del team giallo, con un tempo di attesa iniziale più lungo. Tieni presente che i tempi di attesa nelle espansioni del set di regole sono assoluti, non composti. Quindi la prima espansione avviene a cinque secondi e la seconda espansione avviene cinque secondi dopo, a dieci secondi.

```
"expansions": [{
  "target": "teams[Yellow].minPlayers",
  "steps": [{
    "waitTimeSeconds": 5,
    "value": 8
  }, {
    "waitTimeSeconds": 10,
    "value": 5
  }]
}]
```

Tutorial: Crea un set di regole di matchmaking

Prima di creare un set di regole di matchmaking per il tuo Amazon GameLift Servers FlexMatch matchmaker, ti consigliamo di controllare la sintassi del [set di regole](#). Dopo aver creato un set di regole utilizzando il Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI), non puoi cambiarlo.

Tieni presente che esiste una [quota di servizio](#) per il numero massimo di set di regole che puoi avere in una AWS regione, quindi è una buona idea eliminare i set di regole non utilizzati.

Console

Crea un set di regole

1. Aprire il Amazon GameLift Servers console all'indirizzo <https://console.aws.amazon.com/gamelift/>.
2. Passa alla AWS regione in cui desideri creare il set di regole. Definisci i set di regole nella stessa regione della configurazione di matchmaking che li utilizza.
3. Nel riquadro di navigazione, scegli FlexMatch, Set di regole di matchmaking.
4. Nella pagina Set di regole di matchmaking, scegli Crea set di regole.
5. Nella pagina Crea set di regole di matchmaking, procedi come segue:
 - a. In Impostazioni del set di regole, in Nome, inserisci un nome descrittivo univoco che puoi utilizzare per identificarlo in un elenco o nelle tabelle degli eventi e delle metriche.
 - b. Per Set di regole, inserisci il set di regole in JSON. Per informazioni sulla progettazione di un set di regole, consulta [Progetta un FlexMatch set di regole](#). È inoltre possibile utilizzare uno dei set di regole di esempio di [FlexMatch esempi di set di regole](#).
 - c. Scegli Convalida per verificare che la sintassi del set di regole sia corretta. Non puoi modificare i set di regole dopo averli creati, quindi è una buona idea convalidarli prima.
 - d. (Facoltativo) In Tag, aggiungi tag per aiutarti a gestire e tenere traccia AWS delle tue risorse.
6. Scegli Create (Crea) . Se la creazione ha esito positivo, puoi utilizzare la regola impostata con un matchmaker.

AWS CLI

Crea un set di regole

Aprire una finestra della riga di comando e utilizzare il comando [create-matchmaking-rule-set](#).

Questo comando di esempio crea un semplice set di regole di matchmaking che configura una singola squadra. Assicurati di creare il set di regole nella stessa AWS regione delle configurazioni di matchmaking che lo utilizzano.

```
aws gamelift create-matchmaking-rule-set \  
  --name "SampleRuleSet123" \  
  --
```

```
--rule-set-body '{"name": "aliens_vs_cowboys", "ruleLanguageVersion": "1.0",  
"teams": [{"name": "cowboys", "maxPlayers": 8, "minPlayers": 4}]}'
```

Se la richiesta di creazione ha esito positivo, Amazon GameLift Servers restituisce un [MatchmakingRuleSet](#) oggetto che include le impostazioni specificate. Un matchmaker può ora utilizzare il nuovo set di regole.

Console

Eliminazione di un set di regole

1. Apri il Amazon GameLift Servers console all'indirizzo <https://console.aws.amazon.com/gamelift/>.
2. Passa alla regione in cui hai creato la regola impostata.
3. Nel riquadro di navigazione, scegli FlexMatch, Set di regole di matchmaking.
4. Nella pagina Set di regole di matchmaking, seleziona il set di regole che desideri eliminare, quindi scegli Elimina.
5. Nella finestra di dialogo Elimina set di regole, scegli Elimina per confermare l'eliminazione.

Note

Se una configurazione di matchmaking utilizza il set di regole, Amazon GameLift Servers visualizza un messaggio di errore (Impossibile eliminare il set di regole). In tal caso, modifica la configurazione del matchmaking per utilizzare un set di regole diverso, quindi riprova. Per scoprire quali configurazioni di matchmaking utilizzano un set di regole, scegli il nome di un set di regole per visualizzarne la pagina dei dettagli.

AWS CLI

Eliminare un set di regole

Apri una finestra della riga di comando e usa il comando [delete-matchmaking-rule-set](#) per eliminare un set di regole di matchmaking.

Se una configurazione di matchmaking utilizza il set di regole, Amazon GameLift Servers restituisce un messaggio di errore. Se ciò accade, modifica la configurazione del matchmaking

per utilizzare un set di regole diverso, quindi riprova. Per ottenere un elenco delle configurazioni di matchmaking che utilizzano un set di regole, usa il comando [describe-matchmaking-configurationse](#) specifica il nome del set di regole.

Questo comando di esempio verifica l'utilizzo del set di regole di matchmaking e quindi elimina il set di regole.

```
aws gamelift describe-matchmaking-rule-sets \  
  --rule-set-name "SampleRuleSet123" \  
  --limit 10  
  
aws gamelift delete-matchmaking-rule-set \  
  --name "SampleRuleSet123"
```

FlexMatch esempi di set di regole

FlexMatch i set di regole possono coprire una varietà di scenari di matchmaking. I seguenti esempi sono conformi al FlexMatch struttura di configurazione e linguaggio di espressione delle proprietà. Copiare questi set di regole interamente o scegliere i componenti in base alle esigenze.

Per ulteriori informazioni sull'utilizzo FlexMatch regole e set di regole, vedere i seguenti argomenti:

Note

Durante la valutazione di un ticket di abbinamento che include più giocatori, tutti i giocatori nella richiesta devono soddisfare i requisiti di corrispondenza.

Argomenti

- [Esempio: crea due squadre con giocatori allo stesso livello](#)
- [Esempio: crea squadre irregolari \(Hunters vs Monster\)](#)
- [Esempio: imposta i requisiti a livello di team e i limiti di latenza](#)
- [Esempio: utilizza l'ordinamento esplicito per trovare le migliori corrispondenze](#)
- [Esempio: trova le intersezioni tra gli attributi di più giocatori](#)
- [Esempio: confronta gli attributi di tutti i giocatori](#)
- [Esempio: crea una partita di grandi dimensioni](#)
- [Esempio: crea una grande partita a più squadre](#)

- [Esempio: crea una partita di grandi dimensioni con giocatori con caratteristiche simili](#)
- [Esempio: usa una regola composta per creare una partita con giocatori con attributi simili o selezioni simili](#)
- [Esempio: crea una regola che utilizza la lista dei blocchi di un giocatore](#)

Esempio: crea due squadre con giocatori allo stesso livello

Questo esempio illustra come impostare due squadre ugualmente abbinate di giocatori con le seguenti istruzioni.

- Creare due squadre di giocatori.
 - Includere da quattro a otto giocatori in ciascuna squadra.
 - Le squadre finali devono avere lo stesso numero di giocatori.
- Includere il livello di competenza dei giocatori (se non viene fornito, l'impostazione predefinita è su 10).
- Scegliere i giocatori se il loro livello di competenza è analogo ad altri giocatori. Verificare che entrambe le squadre dispongano di una competenza media dei giocatori entro i 10 punti.
- Se la corrispondenza non viene completata in modo rapido, rendere più flessibili i requisiti in termini di competenze dei giocatori per completare una corrispondenza in tempi ragionevoli.
 - Dopo 5 secondi, espandere la ricerca per accettare le squadre con competenze medie dei giocatori entro i 50 punti.
 - Dopo 15 secondi, espandere la ricerca per accettare le squadre con competenze medie dei giocatori entro i 100 punti.

Note sull'utilizzo di questo set di regole:

- Questo esempio consente alle squadre di avere qualsiasi dimensione compresa tra quattro e otto giocatori (anche se devono avere le stesse dimensioni). Per le squadre con un'ampia gamma di dimensioni valide, il matchmaker tenta tutte le soluzioni possibili per abbinare il numero massimo di giocatori consentiti.
- La regola `FairTeamSkill` garantisce che le squadre vengano abbinate in modo uniforme in base alle competenze dei giocatori. Per valutare questa regola per ogni nuovo potenziale giocatore, FlexMatch aggiunge provvisoriamente il giocatore a una squadra e calcola le medie. Se la regola ha esito negativo, il giocatore potenziale non viene aggiunto alla corrispondenza.

- Poiché entrambi i team hanno strutture identiche, è possibile scegliere di creare una sola definizione di team e impostare la quantità dei team su "2". In questo scenario, se hai denominato il team "aliens", ai team verranno assegnati i nomi "aliens_1" e "aliens_2".

```
{
  "name": "aliens_vs_cowboys",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number",
    "default": 10
  }],
  "teams": [{
    "name": "cowboys",
    "maxPlayers": 8,
    "minPlayers": 4
  }, {
    "name": "aliens",
    "maxPlayers": 8,
    "minPlayers": 4
  }],
  "rules": [{
    "name": "FairTeamSkill",
    "description": "The average skill of players in each team is within 10 points from the average skill of all players in the match",
    "type": "distance",
    // get skill values for players in each team and average separately to produce list of two numbers
    "measurements": [ "avg(teams[*].players.attributes[skill])" ],
    // get skill values for players in each team, flatten into a single list, and average to produce an overall average
    "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
    "maxDistance": 10 // minDistance would achieve the opposite result
  }, {
    "name": "EqualTeamSizes",
    "description": "Only launch a game when the number of players in each team matches, e.g. 4v4, 5v5, 6v6, 7v7, 8v8",
    "type": "comparison",
    "measurements": [ "count(teams[cowboys].players)" ],
    "referenceValue": "count(teams[aliens].players)",
    "operation": "=" // other operations: !=, <, <=, >, >=
  }],
}
```



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

Esempio: crea squadre irregolari (Hunters vs Monster)

Questo esempio descrive una modalità di gioco in cui un gruppo di giocatori caccia un singolo mostro. Le persone scelgono il ruolo di cacciatore o di mostro. I cacciatori specificano il livello minimo di competenza per il mostro che desiderano affrontare. La dimensione minima della squadra del cacciatore può diventare più flessibile nel corso del tempo per completare la corrispondenza. Questo scenario delinea le seguenti istruzioni:

- Creazione di una squadra di esattamente cinque cacciatori.
- Creazione di una squadra di esattamente un mostro.
- Includere i seguenti attributi giocatore:
 - Il livello di competenza di un giocatore (se non viene fornito, l'impostazione predefinita è su 10).
 - Il livello di competenza del mostro preferito del giocatore (se non viene fornito, l'impostazione predefinita è su 10).
 - Se il giocatore vuole essere il mostro (se non viene fornita questa informazione, l'impostazione predefinita è su 0 o falso).
- Scegliere un giocatore come mostro in base ai seguenti criteri:
 - Il giocatore deve richiedere il ruolo di mostro.
 - Il giocatore deve soddisfare o superare il livello massimo di competenza preferito dai giocatori che sono già parte della squadra di cacciatori.
- Scegliere i giocatori per la squadra dei cacciatori in base ai seguenti criteri:
 - I giocatori che richiedono il ruolo di mostro non possono partecipare alla squadra dei cacciatori.
 - Se il ruolo di mostro è già stato ricoperto, il giocatore deve cercare un livello di abilità per il mostro inferiore alla capacità del mostro proposto.

- Se una corrispondenza non viene completata in modo rapido, rendere la dimensione minima della squadra dei cacciatori più flessibile come segue:
 - Dopo 30 secondi, consentire al gioco di iniziare con solo quattro giocatori nella squadra cacciatori.
 - Dopo 60 secondi, consentire al gioco di iniziare con solo tre giocatori nella squadra cacciatori.

Note sull'utilizzo di questo set di regole:

- Utilizzando due squadre separate per cacciatori e mostro, è possibile valutare l'iscrizione in base a diversi set di criteri.

```
{
  "name": "players_vs_monster_5_vs_1",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number",
    "default": 10
  }, {
    "name": "desiredSkillOfMonster",
    "type": "number",
    "default": 10
  }, {
    "name": "wantsToBeMonster",
    "type": "number",
    "default": 0
  }],
  "teams": [{
    "name": "players",
    "maxPlayers": 5,
    "minPlayers": 5
  }, {
    "name": "monster",
    "maxPlayers": 1,
    "minPlayers": 1
  }],
  "rules": [{
    "name": "MonsterSelection",
    "description": "Only users that request playing as monster are assigned to the monster team",
  }]
```

```

    "type": "comparison",
    "measurements": ["teams[monster].players.attributes[wantsToBeMonster]"],
    "referenceValue": 1,
    "operation": "="
  },{
    "name": "PlayerSelection",
    "description": "Do not place people who want to be monsters in the players
team",
    "type": "comparison",
    "measurements": ["teams[players].players.attributes[wantsToBeMonster]"],
    "referenceValue": 0,
    "operation": "="
  },{
    "name": "MonsterSkill",
    "description": "Monsters must meet the skill requested by all players",
    "type": "comparison",
    "measurements": ["avg(teams[monster].players.attributes[skill])"],
    "referenceValue":
"max(teams[players].players.attributes[desiredSkillOfMonster])",
    "operation": ">="
  }],
  "expansions": [{
    "target": "teams[players].minPlayers",
    "steps": [{
      "waitTimeSeconds": 30,
      "value": 4
    },{
      "waitTimeSeconds": 60,
      "value": 3
    }]
  }]
}

```

Esempio: imposta i requisiti a livello di team e i limiti di latenza

Questo esempio illustra come impostare team di giocatori e applicare un set di regole a ciascun team anziché a ciascun singolo giocatore. Utilizza una singola definizione per creare tre team ugualmente abbinati. Inoltre, stabilisce una latenza massima per tutti i giocatori. I massimi della latenza possono diventare più flessibili nel corso del tempo per completare la corrispondenza. Questo esempio riporta le seguenti istruzioni:

- Creare tre squadre di giocatori.

- Includere da tre a cinque giocatori in ciascuna squadra.
- le squadre finali devono contenere lo stesso o quasi lo stesso numero di giocatori (all'interno di una stessa squadra).
- Includere i seguente attributi giocatore:
 - Il livello di competenza di un giocatore (se non viene fornito, l'impostazione predefinita è su 10).
 - Il ruolo del carattere del giocatore (se non viene fornito, l'impostazione predefinita è su "peasant").
- Scegliere i giocatori se il loro livello di competenza è analogo ad altri giocatori presenti nella corrispondenza.
 - Verificare che ogni squadra disponga di una competenza media dei giocatori entro i 10 punti.
- Limitare le squadre al seguente numero di caratteri "medic":
 - Una corrispondenza intera può avere al massimo cinque medic.
- Vengono abbinati solo i giocatori che segnalano una latenza massima di 50 millisecondi.
- Se una corrispondenza non viene completata in modo rapido, rendere il requisito relativo alla latenza dei giocatori più flessibile come segue:
 - Dopo 10 secondi, consentire valori per la latenza giocatori fino a 100 ms.
 - Dopo 20 secondi, consentire valori per la latenza giocatori fino a 150 ms.

Note sull'utilizzo di questo set di regole:

- Il set di regole garantisce che le squadre vengano abbinati in modo uniforme in base alle competenze dei giocatori. Per valutare la `FairTeamSkill` regola, FlexMatch aggiunge provvisoriamente il potenziale giocatore a una squadra e calcola l'abilità media dei giocatori della squadra. Quindi la confronta con l'abilità media dei giocatori in entrambi i team. Se la regola ha esito negativo, il giocatore potenziale non viene aggiunto alla corrispondenza.
- La squadra e i requisiti in termini di livello di corrispondenza (numero totale di medic) vengono archiviati attraverso una regola di raccolta. Questo tipo di regola richiede un elenco di attributi di caratteri per tutti i giocatori e controlli rispetto ai conteggi massimi. Utilizzare `flatten` per creare un elenco di tutti i giocatori in tutte le squadre.
- Durante la valutazione in base alla latenza, annotare quanto segue:
 - I dati di latenza vengono forniti nella richiesta di abbinamento come parte dell'oggetto `Player`. Non è un attributo del giocatore, perciò non deve essere elencato come uno.

- Il matchmaker valuta la latenza per regione. Qualsiasi regione con una latenza più elevata rispetto al massimo viene ignorata. Per essere accettato per una corrispondenza, un giocatore deve avere almeno una regione con una latenza inferiore al massimo.
- Se una richiesta di abbinamento omette i dati di latenza di uno o più giocatori, la richiesta viene respinta per tutte le corrispondenze.

```
{
  "name": "three_team_game",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number",
    "default": 10
  }, {
    "name": "character",
    "type": "string_list",
    "default": [ "peasant" ]
  }],
  "teams": [{
    "name": "trio",
    "minPlayers": 3,
    "maxPlayers": 5,
    "quantity": 3
  }],
  "rules": [{
    "name": "FairTeamSkill",
    "description": "The average skill of players in each team is within 10 points
from the average skill of players in the match",
    "type": "distance",
    // get players for each team, and average separately to produce list of 3
    "measurements": [ "avg(teams[*].players.attributes[skill])" ],
    // get players for each team, flatten into a single list, and average to
produce overall average
    "referenceValue": "avg(flatten(teams[*].players.attributes[skill]))",
    "maxDistance": 10 // minDistance would achieve the opposite result
  }, {
    "name": "CloseTeamSizes",
    "description": "Only launch a game when the team sizes are within 1 of each
other. e.g. 3 v 3 v 4 is okay, but not 3 v 5 v 5",
    "type": "distance",
    "measurements": [ "max(count(teams[*].players))" ],
```

```

    "referenceValue": "min(count(teams[*].players))",
    "maxDistance": 1
  }, {
    "name": "OverallMedicLimit",
    "description": "Don't allow more than 5 medics in the game",
    "type": "collection",
    // This is similar to above, but the flatten flattens everything into a single
    // list of characters in the game.
    "measurements": [ "flatten(teams[*].players.attributes[character])"],
    "operation": "contains",
    "referenceValue": "medic",
    "maxCount": 5
  }, {
    "name": "FastConnection",
    "description": "Prefer matches with fast player connections first",
    "type": "latency",
    "maxLatency": 50
  }],
  "expansions": [{
    "target": "rules[FastConnection].maxLatency",
    "steps": [{
      "waitTimeSeconds": 10,
      "value": 100
    }, {
      "waitTimeSeconds": 20,
      "value": 150
    }
  ]
}]
}

```

Esempio: utilizza l'ordinamento esplicito per trovare le migliori corrispondenze

Questo esempio imposta una corrispondenza semplice con due squadre di tre giocatori. Illustra come utilizzare regole di ordinamento esplicite per trovare le migliori corrispondenze possibili, il più rapidamente possibile. Queste regole ordinano tutti i ticket di matchmaking attivi per creare le migliori partite in base a determinati requisiti chiave. Questo esempio è implementato con le seguenti istruzioni:

- Creare due squadre di giocatori.
- Includere esattamente tre giocatori in ciascuna squadra.
- Includere i seguente attributi giocatore:

- Livello di esperienza (se non fornito, il livello predefinito è impostato su 50).
- Modalità di gioco preferite (possono essere presenti più valori) (se non forniti, l'impostazione predefinita è su "coop" e "deathmatch").
- Le mappe di gioco preferite, incluso il nome della mappa e il peso delle preferenze (se non specificato, l'impostazione predefinita è su "defaultMap" con un peso di 100).
- Configurare l'ordinamento preventivo:
 - Ordinare i giocatori in base alle loro preferenze per la stessa mappa di gioco del giocatore ancoraggio. I giocatori possono avere più mappe di gioco preferite, perciò questo esempio utilizza un valore preferenza.
 - Ordinare giocatori in base a quanto il loro livello di esperienza corrisponda al giocatore di ancoraggio. Con questo ordinamento, tutti i giocatori in tutte le squadre avranno livelli di esperienza più vicini possibile.
- Tutti i giocatori in tutte le squadre devono aver selezionato almeno una modalità di gioco in comune.
- Tutti i giocatori in tutte le squadre devono aver selezionato almeno una mappa di gioco in comune.

Note sull'utilizzo di questo set di regole:

- L'ordinamento della mappa di gioco usa un ordinamento assoluto che confronta il valore dell'attributo mapPreference. Poiché è il primo nel set di regole, questo ordinamento viene eseguito per primo.
- L'esperienza utilizza un ordinamento a distanza per confrontare il livello di abilità di un potenziale giocatore con l'abilità del giocatore di ancoraggio.
- Gli ordinamenti vengono eseguiti nell'ordine in cui sono elencati in un set di regole. In questo scenario, i giocatori sono ordinati in base alle preferenze della mappa di gioco, quindi in base al livello di esperienza.

```
{
  "name": "multi_map_game",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "experience",
    "type": "number",
    "default": 50
  }, {
```

```
    "name": "gameMode",
    "type": "string_list",
    "default": [ "deathmatch", "coop" ]
  }, {
    "name": "mapPreference",
    "type": "string_number_map",
    "default": { "defaultMap": 100 }
  }, {
    "name": "acceptableMaps",
    "type": "string_list",
    "default": [ "defaultMap" ]
  }
}],
"teams": [{
  "name": "red",
  "maxPlayers": 3,
  "minPlayers": 3
}, {
  "name": "blue",
  "maxPlayers": 3,
  "minPlayers": 3
}],
"rules": [{
  // We placed this rule first since we want to prioritize players preferring the
  same map
  "name": "MapPreference",
  "description": "Favor grouping players that have the highest map preference
  aligned with the anchor's favorite",
  // This rule is just for sorting potential matches. We sort by the absolute
  value of a field.
  "type": "absoluteSort",
  // Highest values go first
  "sortDirection": "descending",
  // Sort is based on the mapPreference attribute.
  "sortAttribute": "mapPreference",
  // We find the key in the anchor's mapPreference attribute that has the highest
  value.
  // That's the key that we use for all players when sorting.
  "mapKey": "maxValue"
}, {
  // This rule is second because any tie-breakers should be ordered by similar
  experience values
  "name": "ExperienceAffinity",
  "description": "Favor players with similar experience",
```



```

    // This rule is just for sorting potential matches. We sort by the distance
    from the anchor.
    "type": "distanceSort",
    // Lowest distance goes first
    "sortDirection": "ascending",
    "sortAttribute": "experience"
  }, {
    "name": "SharedMode",
    "description": "The players must have at least one game mode in common",
    "type": "collection",
    "operation": "intersection",
    "measurements": [ "flatten(teams[*].players.attributes[gameMode])" ],
    "minCount": 1
  }, {
    "name": "MapOverlap",
    "description": "The players must have at least one map in common",
    "type": "collection",
    "operation": "intersection",
    "measurements": [ "flatten(teams[*].players.attributes[acceptableMaps])" ],
    "minCount": 1
  }
}

```

Esempio: trova le intersezioni tra gli attributi di più giocatori

Questo esempio illustra come utilizzare una regola di raccolta per trovare intersezioni in due o più attributi giocatore. Quando si utilizzano le raccolte, è possibile utilizzare l'operazione `intersection` per un singolo attributo e l'operazione `reference_intersection_count` per più attributi.

Per illustrare questo approccio, questo esempio valuta i giocatori in una corrispondenza in base alle loro preferenze di caratteri. Il gioco di esempio è uno stile "free-for-all" in cui tutti i giocatori di una partita sono avversari. A ogni giocatore viene chiesto di (1) scegliere un carattere per se stessi e (2) scegliere i caratteri contro cui desiderano giocare. Abbiamo bisogno di una regola che garantisca che ogni giocatore in una corrispondenza utilizzi un carattere che sia nell'elenco degli avversari preferiti di tutti gli altri giocatori.

Il set di regole di esempio descrive una corrispondenza con le seguenti caratteristiche:

- Struttura della squadra: una squadra di cinque giocatori
- Attributi dei giocatori:
 - `myCharacter`: il carattere scelto dal giocatore.

- `preferredOpponents`: elenco di caratteri contro cui il giocatore vuole giocare.
- Regole corrispondenza: una corrispondenza potenziale è accettabile se ogni carattere in uso è nell'elenco di avversari preferiti di ogni giocatore.

Per implementare la regola di corrispondenza, questo esempio utilizza una regola di raccolta con i seguenti valori di proprietà:

- Operazione: utilizza `reference_intersection_count` l'operazione per valutare come ogni elenco di stringhe nel valore di misurazione si interseca con l'elenco di stringhe nel valore di riferimento.
- Misurazione: utilizza l'espressione della `flatten` proprietà per creare un elenco di elenchi di stringhe, con ogni elenco di stringhe contenente il valore dell'attributo `myCharacter` di un giocatore.
- Valore di riferimento: utilizza l'espressione della `set_intersection` proprietà per creare un elenco di stringhe di tutti i valori degli attributi `PreferredOpponents` comuni a tutti i giocatori della partita.
- Restrizioni: `minCount` è impostato su 1 per garantire che il personaggio scelto da ogni giocatore (un elenco di stringhe nella misurazione) corrisponda ad almeno uno degli avversari preferiti comuni a tutti i giocatori. (una stringa nel valore di riferimento).
- Espansione: se una partita non viene riempita entro 15 secondi, allenta il requisito minimo di intersezione.

Il flusso di elaborazione per questa regola è il seguente:

1. Un giocatore viene aggiunto alla corrispondenza potenziale. Il valore di riferimento (un elenco di stringhe) è ricalcolato per includere le intersezioni con l'elenco degli avversari preferiti del nuovo giocatore. Il valore di misurazione (un elenco di elenchi di stringhe) viene ricalcolato per aggiungere il carattere scelto dal nuovo giocatore come un nuovo elenco stringa.
2. Amazon GameLift Servers verifica che ogni elenco di stringhe nel valore di misurazione (i caratteri scelti dai giocatori) si intersechi con almeno una stringa nel valore di riferimento (gli avversari preferiti dai giocatori). Poiché in questo esempio ciascun elenco di stringhe nella misurazione contiene un solo valore, l'intersezione può essere 0 o 1.
3. Se qualsiasi elenco di stringhe nella misurazione non si interseca con l'elenco di stringhe del valore di riferimento, la regola ha esito negativo e il nuovo giocatore viene rimosso dalla corrispondenza potenziale.

4. Se una corrispondenza non viene completata entro 15 secondi, annullare il requisito di corrispondenza dell'avversario per completare gli slot giocatore rimanenti nella corrispondenza.

```
{
  "name": "preferred_characters",
  "ruleLanguageVersion": "1.0",

  "playerAttributes": [{
    "name": "myCharacter",
    "type": "string_list"
  }, {
    "name": "preferredOpponents",
    "type": "string_list"
  }],

  "teams": [{
    "name": "red",
    "minPlayers": 5,
    "maxPlayers": 5
  }],

  "rules": [{
    "description": "Make sure that all players in the match are using a character
that is on all other players' preferred opponents list.",
    "name": "OpponentMatch",
    "type": "collection",
    "operation": "reference_intersection_count",
    "measurements": ["flatten(teams[*].players.attributes[myCharacter])"],
    "referenceValue":
"set_intersection(flatten(teams[*].players.attributes[preferredOpponents])",
    "minCount":1
  }],
  "expansions": [{
    "target": "rules[OpponentMatch].minCount",
    "steps": [{
      "waitTimeSeconds": 15,
      "value": 0
    }]
  }]
}
```

Esempio: confronta gli attributi di tutti i giocatori

Questo esempio illustra come confrontare gli attributi giocatore all'interno di un gruppo di giocatori.

Il set di regole di esempio descrive una corrispondenza con le seguenti caratteristiche:

- Struttura del team: due team da un giocatore
- Attributi dei giocatori:
 - `gameMode`: tipo di gioco scelto dal giocatore (se non fornito, l'impostazione predefinita è su "turn-based").
 - `gameMap`: il mondo di gioco scelto dal giocatore (se non fornito, l'impostazione predefinita è su 1).
 - `character`: personaggio scelto dal giocatore (nessun valore predefinito significa che i giocatori devono specificare un personaggio).
- Regole di corrispondenza: i giocatori abbinati devono soddisfare i seguenti requisiti:
 - I giocatori devono scegliere la stessa modalità di gioco.
 - I giocatori devono scegliere la stessa mappa di gioco.
 - I giocatori devono scegliere caratteri diversi.

Note sull'utilizzo di questo set di regole:

- Per implementare la regola di corrispondenza, questo esempio utilizza regole di confronto per verificare i valori degli attributi di tutti i giocatori. Per la modalità e la mappa di gioco, la regola verifica che i valori siano gli stessi. Per il carattere, la regola verifica che i valori siano diversi.
- Questo esempio usa una definizione di giocatore con una proprietà di quantità per creare entrambi i team di giocatori. Ai team sono assegnati i nomi seguenti: "player_1" e "player_2".

```
{
  "name": "",
  "ruleLanguageVersion": "1.0",

  "playerAttributes": [{
    "name": "gameMode",
    "type": "string",
    "default": "turn-based"
  }, {
    "name": "gameMap",
```

```
    "type": "number",
    "default": 1
  }, {
    "name": "character",
    "type": "number"
  }],

  "teams": [{
    "name": "player",
    "minPlayers": 1,
    "maxPlayers": 1,
    "quantity": 2
  }],

  "rules": [{
    "name": "SameGameMode",
    "description": "Only match players when they choose the same game type",
    "type": "comparison",
    "operation": "=",
    "measurements": ["flatten(teams[*].players.attributes[gameMode])"]
  }, {
    "name": "SameGameMap",
    "description": "Only match players when they're in the same map",
    "type": "comparison",
    "operation": "=",
    "measurements": ["flatten(teams[*].players.attributes[gameMap])"]
  }, {
    "name": "DifferentCharacter",
    "description": "Only match players when they're using different characters",
    "type": "comparison",
    "operation": "!=",
    "measurements": ["flatten(teams[*].players.attributes[character])"]
  }
  ]
}
```

Esempio: crea una partita di grandi dimensioni

Questo esempio illustra come impostare un set di regole per gli abbinamenti che possono superare i 40 giocatori. Quando un set di regole descrive i team con un numero di maxPlayer maggiore di 40, viene elaborato come abbinamento grande. Ulteriori informazioni in [Progetta un FlexMatch set di regole per partite di grandi dimensioni](#).

Il set di regole di esempio crea un abbinamento utilizzando le seguenti istruzioni:

- Crea un team con un massimo di 200 giocatori, con un requisito minimo di 175 giocatori.
- Criteri di bilanciamento: seleziona i giocatori in base al livello di abilità simile. Tutti i giocatori devono segnalare il livello di abilità da abbinare.
- Preferenza di raggruppamento: raggruppa i giocatori con criteri di bilanciamento simili durante la creazione degli abbinamenti.
- Regole di latenza: imposta la latenza massima accettabile per il giocatore di 150 millisecondi.
- Se l'abbinamento non viene completato rapidamente, rendi più flessibili i requisiti per completare un abbinamento in tempi ragionevoli.
 - Dopo 10 secondi, accetta un team con 150 giocatori.
 - Dopo 12 secondi, aumenta il numero massimo di latenza accettabile a 200 millisecondi.
 - Dopo 15 secondi, accetta un team con 100 giocatori.

Note sull'utilizzo di questo set di regole:

- Poiché l'algoritmo utilizza la preferenza di raggruppamento "popolazione più ampia", i giocatori vengono prima ordinati in base ai criteri di bilanciamento. Di conseguenza, gli abbinamenti tendono ad essere più complete e contengono giocatori che hanno abilità simili. Tutti i giocatori soddisfano i requisiti di latenza accettabili, ma potrebbero non ottenere la migliore latenza possibile per la loro posizione.
- La strategia dell'algoritmo utilizzata in questo set di regole, "popolazione più ampia", è l'impostazione predefinita. Per utilizzare l'impostazione predefinita, è possibile decidere di omettere l'impostazione.
- Se hai attivato il backfill degli abbinamenti, non rendere più flessibili i requisiti di numero dei giocatori troppo rapidamente o potresti avere troppe sessioni di gioco parzialmente completate. Ulteriori informazioni in [Allenta i requisiti delle partite di grandi dimensioni](#).

```
{
  "name": "free-for-all",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "skill",
    "type": "number"
  }],
  "algorithm": {
    "balancedAttribute": "skill",
    "strategy": "balanced",
```

```
    "batchingPreference": "largestPopulation"
  },
  "teams": [{
    "name": "Marauders",
    "maxPlayers": 200,
    "minPlayers": 175
  }],
  "rules": [{
    "name": "low-latency",
    "description": "Sets maximum acceptable latency",
    "type": "latency",
    "maxLatency": 150
  }],
  "expansions": [{
    "target": "rules[low-latency].maxLatency",
    "steps": [{
      "waitTimeSeconds": 12,
      "value": 200
    }],
  }, {
    "target": "teams[Marauders].minPlayers",
    "steps": [{
      "waitTimeSeconds": 10,
      "value": 150
    }, {
      "waitTimeSeconds": 15,
      "value": 100
    }],
  }],
  }]
```

Esempio: crea una grande partita a più squadre

Questo esempio illustra come impostare un set di regole per gli abbinamenti con più team che possono superare i 40 giocatori. Questo esempio illustra come creare più team identici con una definizione e come vengono completati i team di dimensioni asimmetriche durante la creazione dell'abbinamento.

Il set di regole di esempio crea un abbinamento utilizzando le seguenti istruzioni:

- Crea dieci team di "cacciatori" identici con un massimo di 15 giocatori e un team di "mostri" con esattamente 5 giocatori.

- Criteri di bilanciamento: seleziona i giocatori in base al numero di abilità dei mostri. Se i giocatori non segnalano il numero di uccisioni, utilizza un valore predefinito di 5.
- Preferenza di raggruppamento: raggruppa i giocatori in base alle regioni che segnalano la latenza del giocatore più veloce.
- Regola di latenza: imposta una latenza massima accettabile per il giocatore di 200 millisecondi.
- Se l'abbinamento non viene completato rapidamente, rendi più flessibili i requisiti per completare un abbinamento in tempi ragionevoli.
 - Dopo 15 secondi, accetta i team con 10 giocatori.
 - Dopo 20 secondi, accetta i team con 8 giocatori.

Note sull'utilizzo di questo set di regole:

- Questo set di regole definisce le squadre che possono contenere fino a 155 giocatori, il che la rende una partita di grandi dimensioni. (10 x 15 cacciatori+5 mostri = 155)
- Poiché l'algoritmo utilizza la preferenza di raggruppamento "regione più veloce", i giocatori tendono a essere collocati in regioni in cui segnalano una latenza più rapida e non in regioni in cui presentano una latenza elevata (ma accettabile). Allo stesso tempo, è probabile che gli abbinamenti abbiano meno giocatori e che i criteri di bilanciamento (numero di abilità dei mostri) possano variare in modo più ampio.
- Quando viene definita un'espansione per una definizione per più team (quantità > 1), l'espansione si applica a tutti i team creati con quella definizione. Quindi, rendendo più flessibile l'impostazione minima dei giocatori del team di cacciatori, tutti e dieci i team di cacciatori sono ugualmente interessati.
- Poiché questo set di regole è ottimizzato per ridurre al minimo la latenza del giocatore, la regola di latenza funge da catch-all per escludere i giocatori che non hanno opzioni di connessione accettabili. Non abbiamo bisogno di rendere più flessibile questo requisito.
- Ecco come FlexMatch riempie i valori corrispondenti a questo set di regole prima che le espansioni abbiano effetto:
 - Nessun team ha raggiunto il numero di minPlayers ancora. I team di cacciatori dispongono di 15 slot aperti, mentre il team di mostri ha 5 slot aperti.
 - I primi 100 giocatori vengono assegnati (10 ciascuno) ai dieci team di cacciatori.
 - I successivi 22 giocatori vengono assegnati sequenzialmente (2 ciascuno) ai team di cacciatore e al team di mostri.

- I team di cacciatori hanno raggiunto il numero di minPlayers di 12 giocatori ciascuno. Il team di mostri ha 2 giocatori e non ha raggiunto il numero di minPlayers.
 - I successivi tre giocatori vengono assegnati al team di mostri.
- Tutti i team hanno raggiunto il numero di minPlayers. I team di cacciatori hanno tre slot aperti. Il team di mostri è pieno.
 - Gli ultimi 30 giocatori vengono assegnati in sequenza ai team di cacciatori, garantendo che tutti i team di cacciatori abbiano quasi le stesse dimensioni (più o meno un giocatore).
- Se hai attivato il backfill per gli abbinamenti creati con questo set di regole, non rendere più flessibili i requisiti di numero dei giocatori troppo rapidamente o potresti avere troppe sessioni di gioco parzialmente completate. Ulteriori informazioni in [Allenta i requisiti delle partite di grandi dimensioni](#).

```
{
  "name": "monster-hunters",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "monster-kills",
    "type": "number",
    "default": 5
  }],
  "algorithm": {
    "balancedAttribute": "monster-kills",
    "strategy": "balanced",
    "batchingPreference": "fastestRegion"
  },
  "teams": [{
    "name": "Monsters",
    "maxPlayers": 5,
    "minPlayers": 5
  }, {
    "name": "Hunters",
    "maxPlayers": 15,
    "minPlayers": 12,
    "quantity": 10
  }],
  "rules": [{
    "name": "latency-catchall",
    "description": "Sets maximum acceptable latency",
    "type": "latency",
```

```

        "maxLatency": 150
    }],
    "expansions": [{
        "target": "teams[Hunters].minPlayers",
        "steps": [{
            "waitTimeSeconds": 15,
            "value": 10
        }, {
            "waitTimeSeconds": 20,
            "value": 8
        }]
    }]
}

```

Esempio: crea una partita di grandi dimensioni con giocatori con caratteristiche simili

Questo esempio illustra come impostare un set di regole per le partite con due squadre che utilizzano `batchDistance`. Nell'esempio:

- La `SimilarLeague` regola assicura che tutti i giocatori di una partita abbiano un `league` limite di 2 rispetto agli altri giocatori.
- La `SimilarSkill` regola garantisce che tutti i giocatori di una partita abbiano un `skill` limite di 10 punti dagli altri giocatori. Se un giocatore ha aspettato 10 secondi, la distanza viene aumentata a 20. Se un giocatore ha aspettato 20 secondi, la distanza viene aumentata a 40.
- La `SameMap` regola garantisce che tutti i giocatori di una partita abbiano richiesto la stessa `cosamap`.
- La `SameMode` regola garantisce che tutti i giocatori di una partita abbiano richiesto la stessa `cosamode`.

```

{
  "ruleLanguageVersion": "1.0",
  "teams": [{
    "name": "red",
    "minPlayers": 100,
    "maxPlayers": 100
  }, {
    "name": "blue",
    "minPlayers": 100,
    "maxPlayers": 100
  }],
}

```

```
"algorithm": {
  "strategy": "balanced",
  "balancedAttribute": "skill",
  "batchingPreference": "fastestRegion"
},
"playerAttributes": [{
  "name": "league",
  "type": "number"
},{
  "name": "skill",
  "type": "number"
},{
  "name": "map",
  "type": "string"
},{
  "name": "mode",
  "type": "string"
}],
"rules": [{
  "name": "SimilarLeague",
  "type": "batchDistance",
  "batchAttribute": "league",
  "maxDistance": 2
}, {
  "name": "SimilarSkill",
  "type": "batchDistance",
  "batchAttribute": "skill",
  "maxDistance": 10
}, {
  "name": "SameMap",
  "type": "batchDistance",
  "batchAttribute": "map"
}, {
  "name": "SameMode",
  "type": "batchDistance",
  "batchAttribute": "mode"
}],
"expansions": [{
  "target": "rules[SimilarSkill].maxDistance",
  "steps": [{
    "waitTimeSeconds": 10,
    "value": 20
  }, {
    "waitTimeSeconds": 20,
```

```
        "value": 40
      ]
    ]
  }
```

Esempio: usa una regola composta per creare una partita con giocatori con attributi simili o selezioni simili

Questo esempio illustra come impostare un set di regole per le partite con due squadre che utilizzano. compound Nell'esempio:

- La `SimilarLeagueDistance` regola assicura che tutti i giocatori di una partita abbiano un league limite di 2 rispetto agli altri giocatori.
- La `SimilarSkillDistance` regola garantisce che tutti i giocatori di una partita abbiano un skill limite di 10 punti dagli altri giocatori. Se un giocatore ha aspettato 10 secondi, la distanza viene aumentata a 20. Se un giocatore ha aspettato 20 secondi, la distanza viene aumentata a 40.
- La `SameMapComparison` regola garantisce che tutti i giocatori di una partita abbiano richiesto la stessa cosamap.
- La `SameModeComparison` regola garantisce che tutti i giocatori di una partita abbiano richiesto la stessa cosamode.
- La `CompoundRuleMatchmaker` regola garantisce una partita se almeno una delle seguenti condizioni è vera:
 - I giocatori di una partita hanno richiesto la stessa map e la stessa cosamode.
 - I giocatori di una partita hanno league caratteristiche skill e caratteristiche simili.

```
{
  "ruleLanguageVersion": "1.0",
  "teams": [{
    "name": "red",
    "minPlayers": 10,
    "maxPlayers": 20
  }, {
    "name": "blue",
    "minPlayers": 10,
    "maxPlayers": 20
  }],
  "algorithm": {
    "strategy": "balanced",
```

```

    "balancedAttribute": "skill",
    "batchingPreference": "fastestRegion"
  },
  "playerAttributes": [{
    "name": "league",
    "type": "number"
  }, {
    "name": "skill",
    "type": "number"
  }, {
    "name": "map",
    "type": "string"
  }, {
    "name": "mode",
    "type": "string"
  }
  ],
  "rules": [{
    "name": "SimilarLeagueDistance",
    "type": "distance",
    "measurements": ["max(flatten(teams[*].players.attributes[league]))"],
    "referenceValue": "min(flatten(teams[*].players.attributes[league]))",
    "maxDistance": 2
  }, {
    "name": "SimilarSkillDistance",
    "type": "distance",
    "measurements": ["max(flatten(teams[*].players.attributes[skill]))"],
    "referenceValue": "min(flatten(teams[*].players.attributes[skill]))",
    "maxDistance": 10
  }, {
    "name": "SameMapComparison",
    "type": "comparison",
    "operation": "=",
    "measurements": ["flatten(teams[*].players.attributes[map])"]
  }, {
    "name": "SameModeComparison",
    "type": "comparison",
    "operation": "=",
    "measurements": ["flatten(teams[*].players.attributes[mode])"]
  }, {
    "name": "CompoundRuleMatchmaker",
    "type": "compound",
    "statement": "or(and(SameMapComparison, SameModeComparison),
and(SimilarSkillDistance, SimilarLeagueDistance))"
  }
  ],

```

```
"expansions": [{
  "target": "rules[SimilarSkillDistance].maxDistance",
  "steps": [{
    "waitTimeSeconds": 10,
    "value": 20
  }, {
    "waitTimeSeconds": 20,
    "value": 40
  }]
}]
}
```

Esempio: crea una regola che utilizza la lista dei blocchi di un giocatore

Questo esempio illustra un set di regole che consente ai giocatori di evitare di essere abbinati a determinati altri giocatori. I giocatori possono creare una lista di blocco, che il matchmaker valuta durante la selezione dei giocatori per una partita. Per ulteriori informazioni su come aggiungere una lista di blocco o evitare la funzione di elenco, consulta [Games AWS Blog](#).

Questo esempio riporta le seguenti istruzioni:

- Crea due squadre composte esattamente da cinque giocatori.
- Inserisci la lista di blocco di un giocatore, che è una lista di giocatori IDs (fino a 100).
- Confronta tutti i giocatori con la lista di blocco di ogni giocatore e rifiuta una partita proposta se IDs ne trovi uno bloccato.

Note sull'utilizzo di questo set di regole:

- Quando si valuta un nuovo giocatore da aggiungere a una partita proposta (o per recuperare un posto in una partita esistente), il giocatore potrebbe essere respinto per uno dei seguenti motivi:
 - Se il nuovo giocatore è in una lista di blocco per tutti i giocatori che sono già stati selezionati per la partita.
 - Se alcuni giocatori già selezionati per la partita sono nella lista di blocco del nuovo giocatore.
- Come mostrato, questo set di regole impedisce di abbinare un giocatore a qualsiasi giocatore nella sua lista di blocco. Puoi modificare questo requisito in una preferenza (chiamata anche lista «da evitare») aggiungendo un'espansione della regola e aumentando il maxCount valore.

```
{
```

```
"name": "Player Block List",
"ruleLanguageVersion": "1.0",
"teams": [{
  "maxPlayers": 5,
  "minPlayers": 5,
  "name": "red"
}, {
  "maxPlayers": 5,
  "minPlayers": 5,
  "name": "blue"
}],
"playerAttributes": [{
  "name": "BlockList",
  "type": "string_list",
  "default": []
}],
"rules": [{
  "name": "PlayerIdNotInBlockList",
  "type": "collection",
  "operation": "reference_intersection_count",
  "measurements": "flatten(teams[*].players.attributes[BlockList])",
  "referenceValue": "flatten(teams[*].players[playerId])",
  "maxCount": 0
}]
}
```

Crea una configurazione di matchmaking

Per configurare un Amazon GameLift Servers FlexMatch matchmaker per elaborare le richieste di matchmaking, creare una configurazione di matchmaking. Usa uno dei due Amazon GameLift Servers console o AWS Command Line Interface (AWS CLI). Per ulteriori informazioni sulla creazione di un matchmaker, vedere [Progetta un FlexMatch para info](#).

Argomenti

- [Tutorial: Crea un matchmaker per Amazon GameLift Servers hosting](#)
- [Tutorial: Crea un matchmaker indipendente FlexMatch](#)
- [Tutorial: Modifica una configurazione di matchmaking](#)

Tutorial: Crea un matchmaker per Amazon GameLift Servers hosting

Prima di creare una configurazione di matchmaking, [crea un set di regole](#) e un Amazon GameLift Servers [coda di sessione di gioco](#) da utilizzare con il matchmaker.

Console

1. Nella [Amazon GameLift Servers console](#), nel riquadro di navigazione, scegli Configurazioni di matchmaking.
2. Passa alla AWS regione in cui desideri creare il tuo matchmaker.
3. Nella pagina delle configurazioni del matchmaking, scegli Crea configurazione matchmaking.
4. Nella pagina Definisci i dettagli della configurazione, sotto Dettagli di configurazione del matchmaking, procedi come segue:
 - a. Per Nome, inserisci il nome di un matchmaker che possa aiutarti a identificarlo in un elenco e nelle metriche. Il nome del matchmaker deve essere unico all'interno della regione. Le richieste di matchmaking identificano quale matchmaker utilizzare in base al nome e alla regione.
 - b. (Facoltativo) Per la descrizione, aggiungi una descrizione per aiutare a identificare il matchmaker.
 - c. Per Set di regole, scegli un set di regole dall'elenco da usare con il matchmaker. L'elenco contiene tutti i set di regole che hai creato nella regione corrente.
 - d. Per FlexMatch modalità, scegli Gestito per Amazon GameLift Servers hosting gestito. Questa modalità richiede FlexMatch per passare le partite riuscite alla coda della sessione di gioco specificata.
 - e. Per AWS Regione, scegli la regione in cui hai configurato la coda delle sessioni di gioco che desideri utilizzare con il matchmaker.
 - f. Per Queue, scegli la coda della sessione di gioco che desideri utilizzare con il matchmaker.
5. Scegli Next (Successivo).
6. Nella pagina Configura impostazioni, in Impostazioni matchmaking, procedi come segue:
 - a. Per Request timeout, imposta il tempo massimo, in secondi, entro il quale il matchmaker può completare una partita per ogni richiesta. FlexMatch annulla le richieste di matchmaking che superano questo tempo.
 - b. Per la modalità Backfill, scegli una modalità per gestire i backfill delle partite.

- Per attivare la funzione di riempimento automatico, scegli Automatico.
 - Per creare una gestione personalizzata delle richieste di riempimento o per non utilizzare la funzione di riempimento, scegli Manuale.
- c. (Facoltativo) Per il numero di giocatori aggiuntivo, imposta il numero di slot da tenere aperti in una partita. FlexMatch può riempire questi slot con giocatori in futuro.
 - d. (Facoltativo) In Opzioni di accettazione della partita, per Accettazione obbligatoria, se desideri che ogni giocatore di una partita proposta accetti attivamente la partecipazione alla partita, seleziona Obbligatorio. Se selezioni questa opzione, in Timeout di accettazione, imposta per quanto tempo, in secondi, vuoi che il matchmaker attenda le accettazioni dei giocatori prima di annullare la partita.
7. (Facoltativo) In Impostazioni di notifica degli eventi, procedi come segue:
 - a. (Facoltativo) Per l'argomento SNS, scegli un argomento Amazon Simple Notification Service (Amazon SNS) per ricevere notifiche sugli eventi di matchmaking. Se non hai ancora impostato un argomento SNS, puoi sceglierlo in un secondo momento modificando la configurazione del matchmaking. Per ulteriori informazioni, consulta [Configurazione FlexMatch notifiche di eventi](#).
 - b. (Facoltativo) Per i dati personalizzati sugli eventi, inserisci tutti i dati personalizzati che desideri associare a questo matchmaker nella messaggistica degli eventi. FlexMatch include questi dati in ogni evento associato al matchmaker.
 8. (Facoltativo) Espandi Dati di gioco aggiuntivi, quindi procedi come segue:
 - a. (Facoltativo) Per i dati della sessione di gioco, inserisci tutte le informazioni aggiuntive relative al gioco che desideri FlexMatch per rispondere a nuove sessioni di gioco iniziate con partite effettuate utilizzando questa configurazione di matchmaking.
 - b. (Facoltativo) Per le proprietà del gioco, aggiungi le proprietà della coppia chiave-valore che contengono informazioni su una nuova sessione di gioco.
 9. (Facoltativo) In Tag, aggiungi tag per aiutarti a gestire e tenere traccia delle tue AWS risorse.
 10. Scegli Next (Successivo).
 11. Nella pagina Rivedi e crea, rivedi le tue scelte, quindi scegli Crea. Dopo una creazione riuscita, il matchmaker è pronto ad accettare le richieste di matchmaking.

AWS CLI

Per creare una configurazione di matchmaking con AWS CLI, apri una finestra della riga di comando e usa il [create-matchmaking-configuration](#) comando per definire un nuovo matchmaker.

Questo comando di esempio crea una nuova configurazione di matchmaking che richiede l'accettazione del giocatore e consente il riempimento automatico. Riserva inoltre due slot per due giocatori per FlexMatch per aggiungere giocatori in un secondo momento e fornisce alcuni dati sulla sessione di gioco.

```
aws gamelift create-matchmaking-configuration \  
  --name "SampleMatchmaker123" \  
  --description "The sample test matchmaker with acceptance" \  
  --flex-match-mode WITH_QUEUE \  
  --game-session-queue-arns "arn:aws:gamelift:us-  
west-2:111122223333:gamesessionqueue/MyGameSessionQueue" \  
  --rule-set-name "MyRuleSet" \  
  --request-timeout-seconds 120 \  
  --acceptance-required \  
  --acceptance-timeout-seconds 30 \  
  --backfill-mode AUTOMATIC \  
  --notification-target "arn:aws:sns:us-  
west-2:111122223333:My_Matchmaking_SNS_Topic" \  
  --additional-player-count 2 \  
  --game-session-data "key=map,value=winter444"
```

Se la richiesta di creazione della configurazione del matchmaking ha esito positivo, Amazon GameLift Servers restituisce un [MatchmakingConfiguration](#) oggetto con le impostazioni che hai richiesto per il matchmaker. Il nuovo matchmaker è pronto ad accettare le richieste di matchmaking.

Tutorial: Crea un matchmaker indipendente FlexMatch

Prima di creare una configurazione di matchmaking, [crea un set di regole](#) da utilizzare con il matchmaker.

Console

1. Apri il Amazon GameLift Servers console a <https://console.aws.amazon.com/gamelift/casa>.

2. Passa alla AWS regione in cui desideri creare il tuo matchmaker. Per un elenco di regioni che supportano FlexMatch configurazioni di matchmaking, vedere. [Scegli una posizione per il matchmaker](#)
3. Nel riquadro di navigazione, scegli FlexMatch, Configurazioni del matchmaking.
4. Nella pagina delle configurazioni del matchmaking, scegli Crea configurazione di matchmaking.
5. Nella pagina Definisci i dettagli della configurazione, sotto Dettagli di configurazione del matchmaking, procedi come segue:
 - a. Per Nome, inserisci il nome di un matchmaker che possa aiutarti a identificarlo in un elenco e nelle metriche. Il nome del matchmaker deve essere unico all'interno della regione. Le richieste di matchmaking identificano quale matchmaker utilizzare in base al nome e alla regione.
 - b. (Facoltativo) Per la descrizione, aggiungi una descrizione per aiutare a identificare il matchmaker.
 - c. Per Set di regole, scegli un set di regole dall'elenco da usare con il matchmaker. L'elenco contiene tutti i set di regole che hai creato nella regione corrente.
 - d. Per FlexMatch modalità, scegli Standalone. Ciò indica che disponi di un meccanismo personalizzato per avviare nuove sessioni di gioco su una soluzione di hosting esterna a Amazon GameLift Servers.
6. Scegli Next (Successivo).
7. Nella pagina Configura impostazioni, in Impostazioni matchmaking, procedi come segue:
 - a. Per Request timeout, imposta il tempo massimo, in secondi, entro il quale il matchmaker può completare una partita per ogni richiesta. Le richieste di matchmaking che superano questo tempo vengono rifiutate.
 - b. (Facoltativo) In Opzioni di accettazione della partita, per Accettazione obbligatoria, se desideri richiedere a ogni giocatore di una partita proposta di accettare attivamente la partecipazione alla partita, seleziona Obbligatorio. Se selezioni questa opzione, in Timeout di accettazione, imposta per quanto tempo, in secondi, vuoi che il matchmaker attenda le accettazioni dei giocatori prima di annullare la partita.
8. (Facoltativo) In Impostazioni di notifica degli eventi, procedi come segue:
 - a. (Facoltativo) Per l'argomento SNS, scegli un argomento Amazon SNS per ricevere notifiche sugli eventi di matchmaking. Se non hai ancora impostato un argomento SNS,

puoi sceglierlo in un secondo momento modificando la configurazione del matchmaking. Per ulteriori informazioni, consulta [Configurazione FlexMatch notifiche di eventi](#).

- b. (Facoltativo) Per i dati personalizzati sugli eventi, inserisci tutti i dati personalizzati che desideri associare a questo matchmaker nella messaggistica degli eventi. FlexMatch include questi dati in ogni evento associato al matchmaker.
9. (Facoltativo) In Tag, aggiungi tag per aiutarti a gestire e tracciare AWS le tue risorse.
10. Scegli Next (Successivo).
11. Nella pagina Rivedi e crea, rivedi le tue scelte, quindi scegli Crea. Dopo una creazione riuscita, il matchmaker è pronto ad accettare le richieste di matchmaking.

AWS CLI

Per creare una configurazione di matchmaking con AWS CLI, apri una finestra della riga di comando e usa il [create-matchmaking-configuration](#) comando per definire un nuovo matchmaker.

Questo comando di esempio crea una nuova configurazione di matchmaking per un matchmaker autonomo che richiede l'accettazione del giocatore.

```
aws gamelift create-matchmaking-configuration \  
  --name "SampleMatchamker123" \  
  --description "The sample test matchmaker with acceptance" \  
  --flex-match-mode STANDALONE \  
  --rule-set-name "MyRuleSetOne" \  
  --request-timeout-seconds 120 \  
  --acceptance-required \  
  --acceptance-timeout-seconds 30 \  
  --notification-target "arn:aws:sns:us-  
west-2:111122223333:My_Matchmaking_SNS_Topic"
```

Se la richiesta di creazione della configurazione del matchmaking ha esito positivo, Amazon GameLift Servers restituisce un [MatchmakingConfiguration](#) oggetto con le impostazioni che hai richiesto per il matchmaker. Il nuovo matchmaker è pronto ad accettare le richieste di matchmaking.

Tutorial: Modifica una configurazione di matchmaking

Per modificare una configurazione di matchmaking, scegli Configurazioni di matchmaking dalla barra di navigazione e scegli la configurazione che desideri modificare. Puoi aggiornare qualsiasi campo in una configurazione esistente tranne il nome.

Quando si aggiorna un set di regole di configurazione, un nuovo set di regole può essere incompatibile se esistono ticket di matchmaking attivi esistenti per i seguenti motivi:

- Nomi o numero di squadre nuovi o diversi
- Nuovi attributi dei giocatori
- Modifiche ai tipi di attributi dei giocatori esistenti

Per apportare una di queste modifiche al set di regole, crea una nuova configurazione di matchmaking con il set di regole aggiornato.

Configurazione FlexMatch notifiche di eventi

Puoi utilizzare le notifiche degli eventi per tenere traccia dello stato delle singole richieste di matchmaking. Tutti i giochi in produzione o in fase di pre-produzione con attività di matchmaking ad alto volume devono utilizzare le notifiche relative agli eventi.

Sono disponibili due opzioni per impostare le notifiche di eventi.

- Chiedi al tuo matchmaker di pubblicare le notifiche degli eventi su un argomento di Amazon Simple Notification Service (Amazon SNS).
- Usa EventBridge gli eventi Amazon pubblicati automaticamente e la sua suite di strumenti per la gestione degli eventi.

Per un elenco dei FlexMatch eventi che Amazon GameLift Servers emette, vedi [FlexMatch eventi di matchmaking](#).

Argomenti

- [Imposta eventi EventBridge](#)
- [Tutorial: configurare un argomento Amazon SNS](#)
- [Imposta un argomento SNS con crittografia lato server](#)
- [Configurare un abbonamento a un argomento per richiamare una funzione Lambda](#)

Imposta eventi EventBridge

Amazon GameLift Servers pubblica automaticamente tutti gli eventi di matchmaking su Amazon EventBridge. Con EventBridge, puoi impostare regole per indirizzare gli eventi di matchmaking agli obiettivi per l'elaborazione. Ad esempio, puoi impostare una regola per indirizzare l'evento "PotentialMatchCreated" a una AWS Lambda funzione che gestisce le accettazioni dei giocatori. Per ulteriori informazioni, consulta [What is Amazon EventBridge?](#)

Note

Quando configuri i tuoi matchmakers, mantieni vuoto il campo di destinazione della notifica o fai riferimento a un argomento SNS se desideri utilizzare entrambi EventBridge e Amazon SNS.

Tutorial: configurare un argomento Amazon SNS

Puoi avere Amazon GameLift Servers pubblica tutti gli eventi che a FlexMatch matchmaker genera un argomento Amazon SNS.

Per creare un argomento SNS per Amazon GameLift Servers notifiche di eventi

1. Apri la [console Amazon SNS](#).
2. Nel pannello di navigazione, scegli Topics (Argomenti).
3. Nella pagina Topics (Argomenti), seleziona Create new topic (Crea nuovo argomento).
4. Creare un argomento nella console . Per ulteriori informazioni, consulta [Creare un argomento utilizzando la Guida per gli AWS Management Console](#) sviluppatori di Amazon Simple Notification Service.
5. Nella pagina dei dettagli del tuo argomento, scegli Modifica.
6. (Facoltativo) Nella pagina Modifica dell'argomento, espandi la politica di accesso, quindi aggiungi la sintassi in grassetto della seguente dichiarazione di politica AWS Identity and Access Management (IAM) alla fine della politica esistente. (L'intera politica è mostrata qui per maggiore chiarezza). Assicurati di utilizzare i dettagli di Amazon Resource Name (ARN) per il tuo argomento SNS e Amazon GameLift Servers configurazione del matchmaking.

```
{  
  "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:matchmakingconfiguration/your_configuration_name"
          }
        }
      }
    ]
  }

```

7. Scegli Save changes (Salva modifiche).

Imposta un argomento SNS con crittografia lato server

È possibile utilizzare la crittografia lato server (SSE) per archiviare dati sensibili in argomenti crittografati. SSE protegge il contenuto dei messaggi negli argomenti di Amazon SNS utilizzando chiavi gestite AWS Key Management Service in AWS KMS(). Per ulteriori informazioni sulla crittografia lato server con Amazon SNS, [consulta Encryption at rest](#) nella Amazon Simple Notification Service Developer Guide.

Per impostare un argomento SNS con crittografia lato server, consulta i seguenti argomenti:

- [Creazione di una chiave nella Guida per](#) gli sviluppatori AWS Key Management Service
- [Abilitare SSE per un argomento nella Guida](#) per gli sviluppatori di Amazon Simple Notification Service

Quando crei la tua chiave KMS, utilizza la seguente politica per le chiavi KMS:

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "gamelift.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn":
        "arn:aws:gamelift:your_region:your_account:matchmakingconfiguration/your_configuration_name"
    },
    "StringEquals": {
      "kms:EncryptionContext:aws:sns:topicArn":
        "arn:aws:sns:your_region:your_account:your_sns_topic_name"
    }
  }
}
```


Configurare un abbonamento a un argomento per richiamare una funzione Lambda

Puoi richiamare una funzione Lambda utilizzando le notifiche di eventi pubblicate sul tuo argomento Amazon SNS. Quando configuri il matchmaker, assicurati di impostare l'obiettivo della notifica sull'ARN del tuo argomento SNS.

Il AWS CloudFormation modello seguente configura una sottoscrizione a un argomento SNS denominato per `MyFlexMatchEventTopic` richiamare una funzione Lambda denominata `FlexMatchEventHandlerLambdaFunction`. Il modello crea una politica di autorizzazioni IAM che consente Amazon GameLift Servers per scrivere sull'argomento SNS. Il modello aggiunge quindi le autorizzazioni per l'argomento SNS per richiamare la funzione Lambda.

```
FlexMatchEventTopic:
  Type: "AWS::SNS::Topic"
  Properties:
    KmsMasterKeyId: alias/aws/sns #Enables server-side encryption on the topic using an
    AWS managed key
    Subscription:
      - Endpoint: !GetAtt FlexMatchEventHandlerLambdaFunction.Arn
        Protocol: lambda
    TopicName: MyFlexMatchEventTopic

FlexMatchEventTopicPolicy:
  Type: "AWS::SNS::TopicPolicy"
  DependsOn: FlexMatchEventTopic
  Properties:
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: Allow
          Principal:
            Service: gamelift.amazonaws.com
          Action:
            - "sns:Publish"
          Resource: !Ref FlexMatchEventTopic
    Topics:
      - Ref: FlexMatchEventTopic

FlexMatchEventHandlerLambdaPermission:
  Type: "AWS::Lambda::Permission"
  Properties:
```

```
Action: "lambda:InvokeFunction"  
FunctionName: !Ref FlexMatchEventHandlerLambdaFunction  
Principal: sns.amazonaws.com  
SourceArn: !Ref FlexMatchEventTopic
```

Preparare il gioco per FlexMatch

Utilizzo Amazon GameLift Servers FlexMatch per aggiungere funzionalità di matchmaking tra giocatori ai tuoi giochi. È possibile utilizzare... FlexMatch con un gestore Amazon GameLift Servers soluzione di hosting o come servizio autonomo con un'altra soluzione di hosting. Se vuoi aggiungere FlexMatch a un Amazon GameLift Servers FleetIQ soluzione, usalo come servizio autonomo. Per ulteriori informazioni su come FlexMatch funziona, vedi [In che modo Amazon GameLift Servers FlexMatch funzionamento](#).

Le soluzioni di matchmaking richiedono il seguente lavoro:

- Crea un matchmaker con le tue regole di matchmaking personalizzate Per ulteriori informazioni sulla creazione del matchmaker, vedere. [Costruire un Amazon GameLift Servers FlexMatch paraninfo](#)
- Aggiorna il tuo client di gioco per consentire ai giocatori di richiedere una partita.
- Per giochi che utilizzano Amazon GameLift Servers hosting, aggiorna il server di gioco per gestire i dati delle partite e, facoltativamente, riempi gli slot vuoti nelle partite.

Gli argomenti di questa sezione spiegano come aggiungere il supporto per il matchmaking ai client e ai server di gioco.

Consulta la tabella di marcia che preferisci FlexMatch soluzione di matchmaking:

- [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#)
- [Tabella di marcia: Crea una soluzione di matchmaking autonoma con FlexMatch](#)

Add (Aggiungi) FlexMatch a un client di gioco

Questo argomento descrive come aggiungere FlexMatch funzionalità di matchmaking per i componenti di gioco lato client.

Consigliamo vivamente al client di gioco di effettuare richieste di matchmaking tramite un servizio di gioco di backend. Utilizzando questa fonte affidabile per la comunicazione con il Amazon GameLift Servers servizio, puoi proteggerti più facilmente dai tentativi di hacking e dai falsi dati dei giocatori. Se il gioco ha un servizio di directory di sessione, questa è una buona opzione per gestire le richieste di abbinamento. Utilizzo di un servizio di gioco di backend per tutte le chiamate verso Amazon GameLift

Servers il servizio è una buona pratica quando si utilizza FlexMatch con Amazon GameLift Servers hosting e come servizio autonomo.

Gli aggiornamenti lato client sono necessari indipendentemente dal fatto che si utilizzi FlexMatch con Amazon GameLift Servers hosting gestito o come servizio autonomo con un'altra soluzione di hosting. Utilizzo dell'API del servizio per Amazon GameLift Servers, che fa parte dell' AWS SDK, aggiungi le seguenti funzionalità:

- Richiedi il matchmaking per uno o più giocatori (obbligatorio). A seconda del set di regole di matchmaking, questa richiesta potrebbe richiedere alcuni dati specifici del giocatore, tra cui gli attributi e la latenza del giocatore.
- Tieni traccia dello stato di una richiesta di matchmaking (obbligatorio). In generale, questa attività richiede l'impostazione della notifica degli eventi.
- Richiedi l'accettazione da parte del giocatore di una partita proposta (opzionale). Questa funzionalità richiede un'interazione aggiuntiva con un giocatore per visualizzare i dettagli della partita e consentirgli di accettare o rifiutare la partita.
- Ottieni informazioni sulla connessione alla sessione di gioco e partecipa al gioco (richiesto). Dopo l'avvio di una sessione di gioco per la nuova partita, recupera le informazioni di connessione per la sessione di gioco e usale per connetterti alla sessione di gioco.

Attività prerequisite sul lato client

Prima di poter aggiungere funzionalità lato client al gioco, devi eseguire queste operazioni:

- Aggiungi l' AWS SDK al tuo servizio di backend. Il tuo servizio di backend utilizza le funzionalità di Amazon GameLift Servers API, che fa parte dell' AWS SDK. Consulta la sezione [Amazon GameLift Servers SDKs per i servizi clienti](#) per saperne di più sull' AWS SDK e scaricare la versione più recente. Per le descrizioni e le funzionalità delle API, consulta [Amazon GameLift Servers FlexMatch Riferimento API \(AWS SDK\)](#).
- Configura un sistema di ticket di matchmaking. Tutte le richieste di matchmaking devono avere un ID di ticket univoco. Crea un meccanismo per generare ticket unici IDs e assegnarli alle richieste corrispondenti. Un ID ticket può utilizzare qualsiasi formato di stringa, fino a un massimo di 128 caratteri.
- Raccogli informazioni sul tuo matchmaker. Ottieni le seguenti informazioni dalla configurazione del matchmaking e dal set di regole.
 - Nome della risorsa di configurazione del matchmaking.

- L'elenco degli attributi del giocatore, definiti nel set di regole.
- Recupera i dati del giocatore. Imposta un modo per ottenere dati pertinenti per ogni giocatore da includere nelle tue richieste di matchmaking. Hai bisogno dell'ID del giocatore e dei valori degli attributi del giocatore. Se il tuo set di regole prevede regole di latenza o desideri utilizzare i dati sulla latenza per organizzare sessioni di gioco, raccogli i dati sulla latenza per ogni posizione geografica in cui è probabile che il giocatore venga inserito in una partita.

Richiedi il matchmaking per i giocatori

Aggiungi codice al servizio di backend del gioco per gestire le richieste di matchmaking a un FlexMatch matchmaker. Il processo di richiesta FlexMatch il matchmaking è identico per i giochi che utilizzano FlexMatch con Amazon GameLift Servers hosting e per giochi che utilizzano FlexMatch come soluzione autonoma.

Per creare una richiesta di matchmaking:

Chiama il Amazon GameLift Servers API [StartMatchmaking](#). Ogni richiesta deve contenere le seguenti informazioni.

Matchmaker

Il nome della configurazione di matchmaking da utilizzare per la richiesta. FlexMatch inserisce ogni richiesta nel pool per il matchmaker specificato, e la richiesta viene elaborata in base a come è configurato il matchmaker. Questo include l'applicazione di un limite di tempo, l'indicazione se richiedere l'accettazione degli abbinamenti ai giocatori, la coda da utilizzare durante il posizionamento di una sessione di gioco risultante e così via. Ulteriori informazioni sui matchmaker e sui set di regole sono disponibili in [Progetta un FlexMatch parainfo](#).

ID del ticket

Un ID ticket univoco assegnato alla richiesta. Tutti gli elementi correlati alla richiesta, tra cui gli eventi e le notifiche, fanno riferimento all'ID ticket.

Dati dei giocatori

L'elenco dei giocatori per cui vuoi creare un abbinamento. Se uno qualsiasi dei giocatori nella richiesta non soddisfa i requisiti di corrispondenza, in base alle regole di corrispondenza e ai minimi di latenza, la richiesta di abbinamento non produrrà mai una corrispondenza con esito positivo. È possibile includere fino a dieci giocatori in una richiesta di abbinamento. Quando ci sono più giocatori in una richiesta, FlexMatch cerca di creare una singola partita e assegnare tutti i

giocatori alla stessa squadra (selezionata casualmente). Se una richiesta contiene troppi giocatori da inserire in una delle squadre di abbinamento, la richiesta non verrà abbinata. Ad esempio, se hai configurato il matchmaker per creare corrispondenze 2v2 (due squadre di due giocatori), non puoi inviare una richiesta di abbinamento contenente più di due giocatori.

Note

Un giocatore (identificato dal proprio ID giocatore) può essere incluso solo in una richiesta di abbinamento attiva alla volta. Quando crei una nuova richiesta per un giocatore, qualsiasi ticket di abbinamento attivo con lo stesso ID giocatore viene automaticamente annullato.

Per ogni giocatore elencato, sono inclusi i seguenti dati:

- ID giocatore: ogni giocatore deve avere un ID giocatore univoco, generato dall'utente. Vedi [Generare un giocatore IDs](#).
- Attributi del giocatore: se il matchmaker in uso richiede gli attributi del giocatore, la richiesta deve fornire tali attributi per ogni giocatore. Gli attributi di giocatore richiesti sono definiti nel set di regole del matchmaker, che specifica anche il tipo di dati per l'attributo. L'attributo di un giocatore è facoltativo solo quando il set di regole specifica un valore predefinito per l'attributo. La richiesta di abbinamento non potrà mai essere completata se non fornisce gli attributi richiesti per tutti i giocatori. Per ulteriori informazioni sul set di regole del matchmaker e sugli attributi di giocatore, consulta [Costruisci un FlexMatch set di regole](#) e [FlexMatch esempi di set di regole](#).
- Latenze dei giocatori: se il matchmaker in uso ha una regola di latenza del giocatore, la richiesta deve riportare la latenza per ogni giocatore. I dati di latenza del giocatore sono un elenco di uno o più valori per giocatore. Rappresenta la latenza che il giocatore prova per le regioni nella coda del matchmaker. Se nella richiesta non sono inclusi valori di latenza per un giocatore, il giocatore non può essere abbinato e la richiesta non riesce.

Per recuperare i dettagli della richiesta di partita

Dopo l'invio di una richiesta di incontro, puoi visualizzare i dettagli della richiesta chiamando [DescribeMatchmaking](#) con l'ID del ticket della richiesta. Questa chiamata restituisce le informazioni sulla richiesta, incluso lo stato corrente. Una volta che una richiesta è stata completata, il ticket contiene anche le informazioni necessarie a un client di gioco per connettersi all'abbinamento.

Per annullare una richiesta di abbinamento

Puoi annullare una richiesta di matchmaking in qualsiasi momento chiamando [StopMatchmaking](#) con l'ID del ticket della richiesta.

Tieni traccia degli eventi di matchmaking

Imposta le notifiche per tenere traccia degli eventi che Amazon GameLift Servers emette per i processi di matchmaking. Puoi configurare le notifiche direttamente, creando un argomento SNS o utilizzando Amazon EventBridge. Per ulteriori informazioni su come impostare le notifiche, consulta [Configurazione FlexMatch notifiche di eventi](#). Una volta configurate le notifiche, aggiungi un listener sul servizio del client per rilevare gli eventi e rispondere in base alle esigenze.

È inoltre consigliabile eseguire il backup delle notifiche eseguendo periodicamente il pooling per gli aggiornamenti dello stato quando passa un periodo di tempo significativo senza notifica. Per ridurre al minimo l'impatto sulle prestazioni dell'abbinamento, assicurati di eseguire il pooling solo dopo aver atteso almeno 30 secondi dopo l'invio del ticket di abbinamento o dopo l'ultima notifica ricevuta.

Recupera un ticket di richiesta di matchmaking, incluso lo stato attuale, chiamando [DescribeMatchmaking](#) con l'ID del ticket della richiesta. Consigliamo di eseguire il polling non più di una volta ogni 10 secondi. Questo approccio è utilizzabile solo in scenari di sviluppo a basso volume.

Note

È consigliabile configurare il gioco con le notifiche degli eventi prima di utilizzare l'abbinamento con volumi elevati, ad esempio con i test del carico di pre-produzione. Tutti i giochi in versione pubblica dovrebbero utilizzare le notifiche indipendentemente dal volume. L'approccio con pooling continuo è appropriato solo per i giochi in sviluppo con un basso utilizzo della funzione di abbinamento.

Richiedi l'accettazione del giocatore

Se stai utilizzando un matchmaker con l'accettazione del giocatore attivata, aggiungi il codice al servizio client per gestire il processo di accettazione del giocatore. Il processo di gestione delle accettazioni da parte dei giocatori è identico per i giochi che utilizzano FlexMatch con Amazon GameLift Servers- hosting gestito e per i giochi che utilizzano FlexMatch come soluzione autonoma.

Richiesta di accettazione del giocatore per un abbinamento proposto:

1. Rileva quando un abbinamento proposto richiede l'accettazione del giocatore. Monitora il ticket di abbinamento per rilevare quando lo stato diventa `REQUIRES_ACCEPTANCE`. Una modifica a questo stato attiva il FlexMatch evento `MatchmakingRequiresAcceptance`.
2. Ottenere le accettazioni da parte di tutti i giocatori. Crea un meccanismo per presentare i dettagli dell'abbinamento proposto ad ogni giocatore nel ticket di abbinamento. I giocatori devono essere in grado di indicare che accettano o rifiutano l'abbinamento proposto. Puoi recuperare i dettagli della partita [DescribeMatchmaking](#) chiamando. I giocatori hanno un tempo limitato per rispondere prima che il matchmaker ritiri l'abbinamento proposto e prosegua.
3. Segnala le risposte dei giocatori a FlexMatch. Segnala le risposte dei giocatori chiamando [AcceptMatch](#) con accetta o rifiuta. Tutti i giocatori in una richiesta di abbinamento devono accettare l'abbinamento affinché possa essere eseguito.
4. Gestire i ticket con accettazioni non riuscite. Una richiesta non riesce quando un giocatore dell'abbinamento proposto rifiuta l'abbinamento o non risponde entro il limite di tempo di accettazione. I biglietti per i giocatori che hanno accettato la partita vengono automaticamente restituiti al pool di biglietti. I ticket per i giocatori che non hanno accettato la partita passano allo stato `FAILURE` e non vengono più elaborati. Per i biglietti con più giocatori, se uno dei giocatori presenti nel ticket non ha accettato la partita, l'intero ticket non va a buon fine.

Connect a una partita

Aggiungi codice al tuo servizio clienti per gestire una partita formata correttamente (stato `COMPLETED` o evento `MatchmakingSucceeded`). Ciò include la notifica ai giocatori dell'abbinamento e la distribuzione delle informazioni di connessione ai client di gioco.

Per giochi che utilizzano Amazon GameLift Servers hosting gestito, quando una richiesta di matchmaking viene soddisfatta con successo, le informazioni di connessione alla sessione di gioco vengono aggiunte al ticket di matchmaking. Recupera un ticket di matchmaking completato chiamando [DescribeMatchmaking](#). Le informazioni sulla connessione includono l'indirizzo IP e la porta della sessione di gioco, nonché un ID sessione del giocatore per ciascun ID giocatore. Ulteriori informazioni in [GameSessionConnectionInfo](#). Il tuo client di gioco può utilizzare queste informazioni per connettersi direttamente alla sessione di gioco della partita. La richiesta di connessione deve includere un ID di sessione del giocatore e un ID giocatore. Questi dati associano il giocatore connesso ai dati delle partite della sessione di gioco, che includono le assegnazioni delle squadre (vedi [GameSession](#)).

Per i giochi che utilizzano altre soluzioni di hosting, tra cui Amazon GameLift Servers FleetIQ, devi creare un meccanismo che consenta ai giocatori della partita di connettersi alla sessione di gioco appropriata.

Esempi di richieste di matchmaking

I seguenti frammenti di codice creano richieste di matchmaking per diversi matchmaker. Come descritto, una richiesta deve fornire gli attributi di giocatore richiesti dal matchmaker in uso, come definito nel set di regole del matchmaker. L'attributo fornito deve utilizzare lo stesso tipo di dati, numero (N) o stringa (S) definiti nel set di regole.

```
# Uses matchmaker for two-team game mode based on player skill level
def start_matchmaking_for_cowboys_vs.aliens(config_name, ticket_id, player_id, skill,
team):
    response = gamelift.start_matchmaking(
        ConfigurationName=config_name,
        Players=[{
            "PlayerAttributes": {
                "skill": {"N": skill}
            },
            "PlayerId": player_id,
            "Team": team
        }],
        TicketId=ticket_id)

# Uses matchmaker for monster hunter game mode based on player skill level
def start_matchmaking_for_players_vs.monster(config_name, ticket_id, player_id, skill,
is_monster):
    response = gamelift.start_matchmaking(
        ConfigurationName=config_name,
        Players=[{
            "PlayerAttributes": {
                "skill": {"N": skill},
                "desiredSkillOfMonster": {"N": skill},
                "wantsToBeMonster": {"N": int(is_monster)}
            },
            "PlayerId": player_id
        }],
        TicketId=ticket_id)

# Uses matchmaker for brawler game mode with latency
```

```
def start_matchmaking_for_three_team_brawler(config_name, ticket_id, player_id, skill,
role):
    response = gamelift.start_matchmaking(
        ConfigurationName=config_name,
        Players=[{
            "PlayerAttributes": {
                "skill": {"N": skill},
                "character": {"S": [role]},
            },
            "PlayerId": player_id,
            "LatencyInMs": { "us-west-2": 20}
        }],
        TicketId=ticket_id)

# Uses matchmaker for multiple game modes and maps based on player experience
def start_matchmaking_for_multi_map(config_name, ticket_id, player_id, skill, maps,
modes):
    response = gamelift.start_matchmaking(
        ConfigurationName=config_name,
        Players=[{
            "PlayerAttributes": {
                "experience": {"N": skill},
                "gameMode": {"SL": modes},
                "mapPreference": {"SL": maps}
            },
            "PlayerId": player_id
        }],
        TicketId=ticket_id)
```

Add (Aggiungi) FlexMatch a un Amazon GameLift Servers-server di gioco ospitato

Quando Amazon GameLift Servers crea una partita, genera una serie di dati sui risultati delle partite che descrivono i dettagli chiave del matchmaking, compresi gli incarichi di squadra. Un server di gioco utilizza questi dati, oltre ad altre informazioni sulla sessione di gioco, quando inizia una nuova sessione di gioco per ospitare la partita.

Per i server di gioco ospitati con Amazon GameLift Servers

Il Amazon GameLift Servers richiede a un processo del server di gioco di avviare una sessione di gioco. Fornisce un [GameSession](#) oggetto che descrive il tipo di sessione di gioco da creare e include informazioni specifiche del giocatore, inclusi i dati delle partite.

Per server di gioco ospitati su altre soluzioni

Dopo aver soddisfatto con successo una richiesta di matchmaking, Amazon GameLift Servers emette un evento che include i risultati della partita. Puoi utilizzare questi dati con la tua soluzione di hosting per avviare una sessione di gioco per la partita.

Informazioni sui dati del matchmaker

I dati delle partite includono le seguenti informazioni:

- Un ID di partita univoco
- L'ID della configurazione di matchmaking utilizzata per creare la partita
- I giocatori selezionati per la partita
- Nomi e incarichi delle squadre
- Valori degli attributi dei giocatori utilizzati per formare la partita. Gli attributi possono anche fornire informazioni che definiscono la configurazione di una sessione di gioco. Ad esempio, il server di gioco potrebbe assegnare personaggi ai giocatori in base agli attributi dei giocatori o scegliere una preferenza della mappa di gioco comune a tutti i giocatori. Oppure il gioco potrebbe sbloccare determinate funzionalità o livelli in base al livello di abilità medio del giocatore.

I dati delle partite non includono la latenza dei giocatori. Se hai bisogno di dati sulla latenza dei giocatori attuali, ad esempio per il backfill delle partite, ti consigliamo di procurarti dati aggiornati.

Note

I dati di Matchmaker specificano l'ARN della configurazione completa del matchmaking, che identifica il nome della configurazione, l'account e la regione. AWS Per ospitare giochi con Amazon GameLift Servers, se utilizzi match backfill, ti serve solo il nome della configurazione. Il nome della configurazione è la stringa che segue «:matchmakingconfiguration/». Nel seguente esempio, il nome della configurazione del matchmaking è "». MyMatchmakerConfig

Questo esempio JSON mostra un tipico set di dati di matchmaker. Descrive una partita a due giocatori, con i giocatori abbinati in base ai punteggi di abilità e al livello più alto raggiunto.

```
{
  "matchId": "1111aaaa-22bb-33cc-44dd-5555eeee66ff",
  "matchmakingConfigurationArn": "arn:aws:gamelift:us-west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig",
  "teams": [
    {
      "name": "attacker",
      "players": [
        {
          "playerId": "4444dddd-55ee-66ff-77aa-8888bbbb99cc",
          "attributes": {
            "skills": {
              "attributeType": "STRING_DOUBLE_MAP",
              "valueAttribute": {"Body": 10.0, "Mind": 12.0, "Heart": 15.0, "Soul": 33.0}
            }
          }
        }
      ]
    }, {
      "name": "defender",
      "players": [
        {
          "playerId": "3333cccc-44dd-55ee-66ff-7777aaaa88bb",
          "attributes": {
            "skills": {
              "attributeType": "STRING_DOUBLE_MAP",
              "valueAttribute": {"Body": 11.0, "Mind": 12.0, "Heart": 11.0, "Soul": 40.0}
            }
          }
        }
      ]
    }
  ]
}
```

Configura un server di gioco per FlexMatch

Server di gioco ospitati con Amazon GameLift Servers deve essere integrato con Amazon GameLift Servers server SDK e dispongono delle funzionalità di base descritte in [Aggiungi Amazon GameLift Servers al tuo server di gioco](#). Questa funzionalità consente l'esecuzione del server di gioco su Amazon GameLift Servers ospitare risorse e comunicare con Amazon GameLift Servers servizio. Le seguenti istruzioni descrivono le attività aggiuntive da eseguire per aggiungere FlexMatch funzionalità.

Da aggiungere FlexMatch al tuo server di gioco

1. Usa i dati di matchmaking quando avvii le sessioni di gioco. Il tuo server di gioco implementa una funzione di callback chiamata `onStartGameSession()`. Dopo aver creato una partita, Amazon GameLift Servers cerca un processo del server di gioco disponibile e chiama questa funzione per richiedergli di avviare una sessione di gioco per la partita. Questa chiamata include un oggetto della sessione di gioco ([GameSession](#)). Il tuo server di gioco utilizza le informazioni sulla sessione di gioco, compresi i dati del matchmaker, per avviare la sessione di gioco. Per maggiori dettagli sull'avvio di una sessione di gioco, vedi [Avviare una sessione di gioco](#). Per ulteriori informazioni sui dati del matchmaker, vedere [Informazioni sui dati del matchmaker](#).
2. Gestire le connessioni dei giocatori. Quando ci si connette a una partita abbinata, un client di gioco fa riferimento a un ID giocatore e a un ID sessione del giocatore (vedi [Convalidare un nuovo](#) giocatore). Configura il tuo server di gioco in modo che utilizzi l'ID giocatore per associare un giocatore in arrivo alle informazioni sul giocatore contenute nei dati del matchmaker. I dati del matchmaker identificano l'incarico di squadra di un giocatore e altre informazioni per rappresentarlo nel gioco.
3. Segnalare quando i giocatori escono dal gioco. Assicurati che il server di gioco chiami l'SDK del server [RemovePlayerSession](#) per segnalare un giocatore abbandonato. Questo passaggio è particolarmente importante se utilizzi FlexMatch riempimento per riempire gli slot vuoti nei giochi esistenti. Scopri di più sull'implementazione FlexMatch riempi. [Riempi i giochi esistenti con FlexMatch](#)
4. Richiedi ai nuovi giocatori di completare le partite esistenti (opzionale). Decidi come vuoi completare le partite in diretta. Se il tuo matchmaker ha la modalità di riempimento impostata su «manuale», potresti voler aggiungere il supporto per il backfill al tuo gioco. Se la modalità di riempimento è impostata su «automatica», potresti aver bisogno di un modo per disattivarla per le singole sessioni di gioco. Ad esempio, dopo che una sessione di gioco raggiunge un determinato punto del gioco, potresti voler interrompere il backfill. Scopri di più su come implementare il match backfill in. [Riempi i giochi esistenti con FlexMatch](#)

Riempi i giochi esistenti con FlexMatch

Match backfill utilizza il tuo FlexMatch meccanismi per trovare nuovi giocatori per le sessioni di gioco abbinata esistenti. Sebbene sia sempre possibile aggiungere giocatori a qualsiasi partita (vedi [Unire un giocatore a una sessione di gioco](#)), il match backfill assicura che i nuovi giocatori soddisfino gli stessi criteri di partita dei giocatori attuali. Inoltre, il backfill degli abbinamenti assegna i nuovi giocatori ai team, gestisce l'accettazione da parte dei giocatori e invia informazioni aggiornate sull'abbinamento al processo del server di gioco. Ulteriori informazioni sul backfill degli abbinamenti sono disponibili in [FlexMatch processo di matchmaking](#).

Note

FlexMatch Il backfill non è attualmente disponibile per i giochi che utilizzano Amazon GameLift Servers In tempo reale.

Esistono due tipi di meccanismi di backfill:

- Abilita il riempimento automatico per riempire le sessioni di gioco che iniziano con un numero di giocatori inferiore al numero massimo consentito. Il riempimento automatico non riempie i giocatori che si uniscono al gioco e poi lo abbandonano.
- Imposta un meccanismo di riempimento manuale per sostituire i giocatori che abbandonano una sessione di gioco in corso. Questo meccanismo deve essere in grado di rilevare uno slot aperto e generare una richiesta di riempimento per riempirlo.

Attiva il riempimento automatico

Con riempimento automatico del fiammifero, Amazon GameLift Servers attiva automaticamente una richiesta di riempimento ogni volta che una sessione di gioco inizia con uno o più slot per giocatori non occupati. Questa caratteristica consente di iniziare i giochi non appena individuato il numero minimo di giocatori abbinati, mentre gli slot rimanenti vengono riempiti in un secondo momento con l'abbinamento di altri giocatori. Puoi interrompere il backfill automatico in qualsiasi momento.

Ad esempio, considera un gioco che può contenere da sei a dieci giocatori. FlexMatch individua inizialmente sei giocatori, forma la partita e inizia una nuova sessione di gioco. Con il backfill automatico, la nuova sessione di gioco può richiedere immediatamente altri quattro giocatori.

A seconda dello stile di gioco, potremmo voler consentire ai nuovi giocatori di unirsi in qualsiasi momento durante la sessione di gioco. In alternativa, puoi interrompere il backfill automatico dopo la fase iniziale di configurazione e prima dell'inizio del gioco.

Per aggiungere il backfill automatico al tuo gioco, effettua i seguenti aggiornamenti per il gioco.

1. Abilita il backfill automatico. Il backfill automatico è gestito in una configurazione di abbinamento. Se abilitato, viene utilizzato con tutte le sessioni di gioco abbinate create con quel matchmaker. Amazon GameLift Servers inizia a generare richieste di riempimento per una sessione di gioco non completa non appena la sessione di gioco viene avviata su un server di gioco.

Per attivare il backfill automatico, apri una configurazione di abbinamento e imposta la modalità di backfill su "AUTOMATICO". Per ulteriori dettagli, consulta [Crea una configurazione di matchmaking](#).

2. Attiva la prioritizzazione del backfill. Personalizza il tuo processo di matchmaking per dare priorità al riempimento delle richieste di riempimento prima di creare nuove partite. Nel set di regole di matchmaking, aggiungi un componente algoritmico e imposta la priorità di riempimento su «alta». Per ulteriori dettagli, consulta [Personalizza l'algoritmo di abbinamento](#).
3. Aggiorna la sessione di gioco con nuovi dati del matchmaker. Amazon GameLift Servers aggiorna il server di gioco con le informazioni sulle partite utilizzando la funzione di callback Server SDK `onUpdateGameSession` (vedi [Inizializzazione del processo del server](#)). Aggiungi il codice al server di gioco per gestire gli oggetti delle sessioni di gioco aggiornate come risultato dell'attività di backfill. Ulteriori informazioni in [Aggiorna i dati delle partite sul server di gioco](#).
4. Disattiva il backfill automatico per una sessione di gioco. È possibile decidere di arrestare il backfill automatico in qualsiasi momento durante una singola sessione di gioco. Per interrompere il backfill automatico, aggiungi il codice al client o al server di gioco per creare il Amazon GameLift Servers Chiamata [StopMatchmaking](#) API. Questa chiamata richiede un ID ticket. Utilizza l'ID ticket di backfill nell'ultima richiesta di backfill. Puoi ottenere queste informazioni dai dati di abbinamento della sessione di gioco, che vengono aggiornati come descritto nel passaggio precedente.

Genera richieste di riempimento manuali da un server di gioco

Puoi avviare manualmente le richieste di match backfill dal processo del server di gioco che ospita la sessione di gioco. Il processo del server contiene la maggior parte delle up-to-date informazioni sui giocatori connessi al gioco e sullo stato degli slot vuoti.

Questo argomento presuppone che tu abbia già creato il necessario FlexMatch componenti e processi di matchmaking aggiunti con successo al tuo server di gioco e a un servizio di gioco lato client. Per maggiori dettagli sulla configurazione FlexMatch, consulta [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#).

Per abilitare il backfill degli abbinamenti per il gioco, aggiungi la funzionalità seguente:

- Invio delle richieste di backfill di abbinamento a un matchmaker e monitoraggio dello stato delle richieste.
- Aggiornamento delle informazioni di abbinamento per la sessione di gioco. Per informazioni, consulta [Aggiorna i dati delle partite sul server di gioco](#).

Come per altre funzionalità del server, un server di gioco utilizza il Amazon GameLift Servers SDK del server. Questo SDK è disponibile in C++ e C#.

Per effettuare le richieste di backfill degli abbinamenti dal server di gioco, completa le attività seguenti.

1. Attivare una richiesta di backfill degli abbinamenti. Solitamente, è consigliabile avviare una richiesta di backfill ogni volta che per un gioco abbinato sono presenti uno o più slot giocatori vuoti. È consigliabile collegare le richieste di backfill a circostanze specifiche, ad esempio ricoprire i ruoli di personaggi importanti o bilanciare i team. È inoltre opportuno limitare l'attività di backfilling in base alla durata della sessione di gioco.
2. Creare una richiesta di backfill. Aggiungi codice per creare e inviare richieste di match backfill a un FlexMatch matchmaker. Le richieste di backfill vengono gestite utilizzando questi server: APIs
 - [StartMatchBackfill\(\)](#)
 - [StopMatchBackfill\(\)](#)

Per creare una richiesta di backfill, richiamare `StartMatchBackfill` con le informazioni seguenti. Per annullare una richiesta di backfill, richiamare `StopMatchBackfill` con l'ID del ticket della richiesta di backfill.

- ID ticket: fornisci un ID del ticket di matchmaking (o scegli di generarli automaticamente). Puoi utilizzare lo stesso meccanismo per assegnare il ticket sia alle richieste di matchmaking che IDs a quelle di backfill. Tali ticket vengono elaborati nello stesso modo.

- **Matchmaker:** identifica quale matchmaker utilizzare per la richiesta di riempimento. In genere, è consigliabile utilizzare lo stesso matchmaker utilizzato per creare l'abbinamento originale. Per questa richiesta è necessario un ARN di configurazione dell'abbinamento. Queste informazioni vengono memorizzate nell'oggetto della sessione di gioco ([GameSession](#)), che è stato fornito al processo del server da Amazon GameLift Servers quando si attiva la sessione di gioco. L'ARN di configurazione dell'abbinamento è incluso nella proprietà `MatchmakerData`.
 - **ARN della sessione di gioco:** identifica la sessione di gioco da riempire. Puoi ottenere l'ARN della sessione di gioco chiamando l'API del server [GetGameSessionId\(\)](#). Durante il processo di abbinamento, i ticket delle nuove richieste non dispongono di un ID della sessione di gioco, a differenza dei ticket delle richieste di backfill. La presenza dell'ID della sessione di gioco consente di distinguere i ticket dei nuovi abbinamenti dai ticket di backfill.
 - **Dati del giocatore:** includi le informazioni sul [giocatore \(Player\)](#) per tutti i giocatori attualmente presenti nella sessione di gioco che stai completando. Queste informazioni consentono al matchmaker di individuare gli abbinamenti di giocatori migliori possibili per i giocatori attualmente nella sessione di gioco. Devi includere l'appartenenza alla squadra per ogni giocatore. Non specificare una squadra se non utilizzi il backfill. Se il server di gioco segnala con precisione lo stato di connessione del giocatore, dovrebbe essere possibile acquisire questi dati come segue:
 1. Il processo del server che ospita la sessione di gioco dovrebbe contenere la maggior parte delle up-to-date informazioni sui giocatori attualmente connessi alla sessione di gioco.
 2. Per ottenere le assegnazioni al giocatore IDs, agli attributi e alla squadra, estrai i dati del giocatore dall'oggetto della sessione di gioco ([GameSession](#)), `MatchmakerData` proprietà (vedi [Informazioni sui dati del matchmaker](#)). I dati del matchmaker includono tutti i giocatori abbinati alla sessione di gioco, pertanto sarà necessario estrarre soltanto i dati relativi ai giocatori attualmente connessi.
 3. Per la latenza dei giocatori, se il matchmaker richiede i dati di latenza, raccogliere nuovi valori di latenza da tutti i giocatori attualmente connessi e includerli in ciascun oggetto `Player`. Se i dati di latenza vengono omessi e sul matchmaker è impostata una regola di latenza, la richiesta non verrà abbinata correttamente. I dati di latenza sono necessari per le richieste di backfill soltanto per la regione in cui si trova attualmente la sessione di gioco. È possibile risalire alla regione della sessione di gioco dalla proprietà `GameSessionId` dell'oggetto `GameSession`; questo valore è un ARN in cui è inclusa la regione.
3. Tieni traccia dello stato di una richiesta di riempimento. Amazon GameLift Servers aggiorna il server di gioco sullo stato delle richieste di riempimento utilizzando la funzione di callback `Server`

SDK `onUpdateGameSession` (vedi [Inizializzare](#) il processo del server). Aggiungi il codice per gestire i messaggi di stato, nonché gli oggetti delle sessioni di gioco aggiornati a seguito di richieste di backfill riuscite, all'indirizzo. [Aggiorna i dati delle partite sul server di gioco](#)

Il matchmaker è in grado di elaborare soltanto una richiesta di backfill degli abbinamenti da una sessione di gioco alla volta. [Se devi annullare una richiesta, chiama `\(\)`. `StopMatchBackfill`](#) Se è necessario modificare una richiesta, richiamare `StopMatchBackfill` e inviare una richiesta aggiornata.

Genera richieste di riempimento manuali da un servizio di backend

È consigliabile inviare le richieste di backfill da un servizio di gioco lato client in alternativa al server di gioco. Per utilizzare questa opzione, il servizio lato client deve disporre dell'accesso ai dati correnti sull'attività della sessione di gioco e sulle connessioni del giocatore; se il gioco utilizza un servizio di directory di sessione, questa potrebbe rappresentare una scelta adeguata.

Questo argomento presuppone che tu abbia già creato il necessario FlexMatch componenti e processi di matchmaking aggiunti con successo al tuo server di gioco e a un servizio di gioco lato client. Per maggiori dettagli sulla configurazione FlexMatch, consulta [Tabella di marcia: Aggiungi il matchmaking a Amazon GameLift Servers soluzione di hosting](#).

Per abilitare il backfill degli abbinamenti per il gioco, aggiungi la funzionalità seguente:

- Invio delle richieste di backfill di abbinamento a un matchmaker e monitoraggio dello stato delle richieste.
- Aggiornamento delle informazioni di abbinamento per la sessione di gioco. Consulta la sezione [Aggiorna i dati delle partite sul server di gioco](#)

Come per altre funzionalità client, un servizio di gioco lato client utilizza l' AWS SDK con Amazon GameLift Servers API. Questo SDK è disponibile in C++, C# e altri linguaggi. Per una descrizione generale del client APIs, vedi Amazon GameLift Servers API Reference, che descrive l'API del servizio per Amazon GameLift Servers azioni e collegamenti a guide di riferimento specifiche per lingua.

Per configurare un servizio di gioco lato client per il backfill dei giochi abbinati, completa le attività seguenti.

1. Attivare una richiesta di backfill. In genere, un gioco avvia una richiesta di backfill ogni volta che sono presenti uno o più slot giocatori vuoti su un gioco abbinato. È consigliabile collegare le richieste di backfill a circostanze specifiche, ad esempio ricoprire i ruoli di personaggi importanti o bilanciare i team. È inoltre opportuno limitare il backfilling in base alla durata della sessione di gioco. Qualunque sia il trigger utilizzato, saranno necessarie almeno le informazioni riportate di seguito. Puoi ottenere queste informazioni dall'oggetto della sessione di gioco ([GameSession](#)) chiamando [DescribeGameSessions](#) con un ID della sessione di gioco.
 - Number of currently empty player slots (Numero di slot giocatori attualmente vuoti). Questo valore può essere calcolato a partire dal limite massimo di giocatori e dal loro attuale numero in una sessione di gioco. Il numero attuale di giocatori viene aggiornato ogni volta che il server di gioco contatta il Amazon GameLift Servers servizio per convalidare la connessione di un nuovo giocatore o per segnalare un giocatore abbandonato.
 - Creation policy (Policy di creazione). Questa impostazione indica se la sessione di gioco attualmente accetta nuovi giocatori.

L'oggetto della sessione di gioco contiene altre informazioni potenzialmente utili, tra cui l'ora di inizio della sessione di gioco, le proprietà di gioco personalizzate e i dati del matchmaker.

2. Creare una richiesta di backfill. Aggiungi codice per creare e inviare richieste di match backfill a un FlexMatch matchmaker. Le richieste di backfill vengono gestite utilizzando questi client: APIs
 - [StartMatchBackfill](#)
 - [StopMatchmaking](#)

Per creare una richiesta di backfill, richiamare `StartMatchBackfill` con le informazioni seguenti. Una richiesta di backfill è simile a una richiesta di abbinamento (consultare [Richiedi il matchmaking per i giocatori](#)), con la differenza che la richiesta di backfill identifica anche la sessione di gioco esistente. Per annullare una richiesta di backfill, richiamare `StopMatchmaking` con l'ID del ticket della richiesta di backfill.

- ID biglietto: fornisci un ID del ticket di matchmaking (o scegli di generarli automaticamente). Puoi utilizzare lo stesso meccanismo per assegnare il ticket sia alle richieste di matchmaking che IDs a quelle di backfill. Tali ticket vengono elaborati nello stesso modo.
- Matchmaker: identifica il nome di una configurazione di matchmaking da utilizzare. In genere, è consigliabile utilizzare per il backfilling lo stesso matchmaker utilizzato per creare

l'abbinamento originale. Queste informazioni si trovano in un oggetto della sessione di gioco ([GameSession](#)), `MatchmakerData` proprietà, nella configurazione del matchmaking ARN. Il valore del nome è la stringa che segue `"/matchmakingconfiguration/"` (ad esempio, nel valore ARN `arn:aws:gamelift:us-west-2:111122223333:matchmakingconfiguration/MM-4v4`, il nome della configurazione di abbinamento è `"MM-4v4"`).

- ARN della sessione di gioco: specifica la sessione di gioco da riempire. Utilizzare la proprietà `GameSessionId` dall'oggetto della sessione di gioco; questo ID utilizza il valore ARN richiesto. I ticket di matchmaking ([MatchmakingTicket](#)) per le richieste di backfill hanno l'ID della sessione di gioco durante l'elaborazione; i ticket per le nuove richieste di matchmaking non ottengono un ID della sessione di gioco finché non viene piazzata la partita; la presenza di `at game session ID` è un modo per distinguere tra i biglietti per le nuove partite e i ticket per i backfill.
- Dati del giocatore: includi le informazioni sul giocatore ([giocatore](#)) per tutti i giocatori in corso nella sessione di gioco che stai completando. Queste informazioni consentono al matchmaker di individuare gli abbinamenti di giocatori migliori possibili per i giocatori attualmente nella sessione di gioco. Devi includere l'appartenenza alla squadra per ogni giocatore. Non specificare una squadra se non utilizzi il backfill. Se il server di gioco segnala con precisione lo stato di connessione del giocatore, dovrebbe essere possibile acquisire questi dati come segue:
 1. Chiama [DescribePlayerSessions\(\)](#) con l'ID della sessione di gioco per scoprire tutti i giocatori che sono attualmente connessi alla sessione di gioco. Ogni sessione giocatore include un ID giocatore. È possibile aggiungere un filtro dello stato per recuperare soltanto le sessioni giocatore attive.
 2. Estrai i dati del giocatore dall'oggetto della sessione di gioco ([GameSession](#)), `MatchmakerData` dalla proprietà (vedi [Informazioni sui dati del matchmaker](#)). Usa il giocatore IDs acquisito nel passaggio precedente per ottenere dati solo per i giocatori attualmente connessi. Dal momento che i dati del matchmaker non vengono aggiornati quando i giocatori abbandonano la sessione di gioco, sarà necessario estrarre soltanto i dati sui giocatori correnti.
 3. Per la latenza dei giocatori, se il matchmaker richiede i dati di latenza, raccogliere nuovi valori di latenza da tutti i giocatori attualmente connessi e includerli nell'oggetto `Player`. Se i dati di latenza vengono omessi e sul matchmaker è impostata una regola di latenza, la richiesta non verrà abbinata correttamente. I dati di latenza sono necessari per le richieste di backfill soltanto per la regione in cui si trova attualmente la sessione di gioco. È possibile

risalire alla regione della sessione di gioco dalla proprietà `GameSessionId` dell'oggetto `GameSession`; questo valore è un ARN in cui è inclusa la regione.

3. Monitorare lo stato della richiesta di backfill. Aggiungere del codice per restare in attesa degli aggiornamenti dello stato del ticket di abbinamento. È possibile utilizzare il meccanismo configurato per monitorare i ticket delle nuove richieste di abbinamento (consultare [Tieni traccia degli eventi di matchmaking](#)) tramite le notifiche eventi (impostazione preferita) o il polling. Anche se non è necessario attivare l'attività di accettazione da parte dei giocatori con le richieste di backfill, e anche se le informazioni sui giocatori vengono aggiornate sul server di gioco, è comunque necessario monitorare lo stato del ticket per gestire gli errori e i nuovi invii delle richieste.

Il matchmaker è in grado di elaborare soltanto una richiesta di backfill degli abbinamenti da una sessione di gioco alla volta. Se è necessario annullare una richiesta, richiamare [StopMatchmaking](#). Se è necessario modificare una richiesta, richiamare `StopMatchmaking` e inviare una richiesta aggiornata.

Una volta che una richiesta di backfill ha esito positivo, il server di gioco riceve un oggetto `GameSession` aggiornato e gestisce le attività richieste per aggiungere nuovi giocatori alla sessione di gioco. Ulteriori informazioni su [Aggiorna i dati delle partite sul server di gioco](#).

Aggiorna i dati delle partite sul server di gioco

Indipendentemente dal modo in cui avvii le richieste di ripristino delle partite nel tuo gioco, il tuo server di gioco deve essere in grado di gestire gli aggiornamenti delle sessioni di gioco Amazon GameLift Servers convegnate a seguito di richieste di match backfill.

Quando Amazon GameLift Servers completa una richiesta di match backfill, con successo o meno, richiama il server di gioco utilizzando la funzione di callback `onUpdateGameSession`. Questa chiamata ha tre parametri di input: un ID del ticket di ripristino della partita, un messaggio di stato e un `GameSession` oggetto contenente la maggior parte dei dati di matchmaking, comprese le informazioni sul giocatore. `up-to-date` È necessario aggiungere il codice seguente al server di gioco come parte dell'integrazione del server di gioco:

1. Implementare la funzione `onUpdateGameSession`. Questa funzione deve essere in grado di gestire i seguenti messaggi di stato (`updateReason`):

- `MATCHMAKING_DATA_UPDATE_UPDATED` - I nuovi giocatori sono stati abbinati con successo alla sessione di gioco. L'oggetto `GameSession` contiene i dati del matchmaker aggiornati, inclusi i dati sui giocatori esistenti e su quelli appena abbinati.
 - `BACKFILL_FAILED` — Il tentativo di ripristino della partita non è riuscito a causa di un errore interno. L'oggetto `GameSession` non viene modificato.
 - `BACKFILL_TIMED_OUT` — Il matchmaker non è riuscito a trovare un backfill match entro il limite di tempo. L'oggetto `GameSession` non viene modificato.
 - `BACKFILL_CANCELLED` — La richiesta di riempimento della partita è stata annullata da una chiamata a `(client)` o `(server)`. `StopMatchmaking` `StopMatchBackfill` L'oggetto `GameSession` non viene modificato.
2. Per gli abbinamenti di backfill corretti, utilizzare i dati del matchmaker aggiornati per gestire i nuovi giocatori che si connettono alla sessione di gioco. Per consentire ai giocatori di iniziare a giocare, è necessario utilizzare almeno le assegnazioni al team per i nuovi giocatori, nonché altri attributi giocatore richiesti.
 3. Nella chiamata del server di gioco a Server SDK action [ProcessReady\(\)](#), aggiungi il nome del metodo di callback come parametro di `processoonUpdateGameSession`.

Sicurezza con FlexMatch

La sicurezza del cloud AWS è la massima priorità. In quanto cliente AWS , puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra te AWS e te. Il [modello di responsabilità condivisa](#) descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. Per saperne di più sui programmi di conformità che si applicano a Amazon GameLift Servers, vedi [AWS i servizi rientranti nell'ambito del programma di conformità](#), .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è responsabile anche di altri fattori, tra cui la sensibilità dei dati, i requisiti aziendali AWS e le normative applicabili.

Per informazioni sulla sicurezza relative a Amazon GameLift Servers, tra cui FlexMatch, vedi [Sicurezza in Amazon GameLift Servers](#). Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo Amazon GameLift Servers. Gli argomenti mostrano come configurare Amazon GameLift Servers per raggiungere i tuoi obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere i Amazon GameLift Servers risorse.

Amazon GameLift Servers FlexMatch riferimento

Questa sezione contiene la documentazione di riferimento per il matchmaking con Amazon GameLift Servers FlexMatch.

Argomenti

- [Amazon GameLift Servers FlexMatch Riferimento API \(AWS SDK\)](#)
- [FlexMatch linguaggio delle regole](#)
- [FlexMatch eventi di matchmaking](#)

Amazon GameLift Servers FlexMatch Riferimento API (AWS SDK)

Questo argomento fornisce un elenco basato sulle attività delle operazioni API per Amazon GameLift Servers FlexMatch. La Amazon GameLift Servers FlexMatch l'API del servizio è inclusa nell' AWS SDK nel namespace. `aws.gamelift` [Scarica l'SDK o visualizza il AWS Amazon GameLift Servers Documentazione](#) di riferimento sull'API.

Amazon GameLift Servers FlexMatch fornisce servizi di matchmaking da utilizzare con giochi ospitati con Amazon GameLift Servers soluzioni di hosting (incluso l'hosting gestito per server di gioco personalizzati o Amazon GameLift Servers Hosting in tempo reale su Amazon EC2 con Amazon GameLift Servers FleetIQ), nonché con altri sistemi di hosting come peer-to-peer primitive di elaborazione locali o cloud. Vedi il [Amazon GameLift Servers Guida](#) per gli sviluppatori per ulteriori informazioni su altri Amazon GameLift Servers opzioni di hosting.

Argomenti

- [Imposta regole e processi di matchmaking](#)
- [Richiedi una partita per uno o più giocatori](#)
- [Linguaggi di programmazione compatibili](#)

Imposta regole e processi di matchmaking

Chiama queste operazioni per creare un FlexMatch matchmaker, configura il processo di matchmaking per il tuo gioco, e definisci una serie di regole personalizzate per la creazione di partite e squadre.

Configurazione di abbinamento

- [CreateMatchmakingConfiguration](#)— Crea una configurazione di matchmaking con istruzioni per valutare gruppi di giocatori e creare squadre di giocatori. Quando si utilizza Amazon GameLift Servers per l'hosting, specifica anche come creare una nuova sessione di gioco per la partita.
- [DescribeMatchmakingConfigurations](#)— Recupera le configurazioni di matchmaking definite a Amazon GameLift Servers regione.
- [UpdateMatchmakingConfiguration](#)- Modifica le impostazioni per la configurazione del matchmaking. coda.
- [DeleteMatchmakingConfiguration](#)— Rimuovi una configurazione di matchmaking dalla regione.

Set di regole di abbinamento

- [CreateMatchmakingRuleSet](#)— Crea una serie di regole da utilizzare durante la ricerca delle partite dei giocatori.
- [DescribeMatchmakingRuleSets](#)— Recupera i set di regole di matchmaking definiti in a Amazon GameLift Servers regione.
- [ValidateMatchmakingRuleSet](#)- Verifica la sintassi per una serie di regole di matchmaking.
- [DeleteMatchmakingRuleSet](#)— Rimuovi un set di regole di matchmaking dalla regione.

Richiedi una partita per uno o più giocatori

Chiama queste operazioni dal tuo servizio client di gioco per gestire le richieste di abbinamento dei giocatori.

- [StartMatchmaking](#)— Richiedi il matchmaking per un giocatore o un gruppo che desidera giocare nella stessa partita.
- [DescribeMatchmaking](#)— Ottieni dettagli su una richiesta di matchmaking, compreso lo stato.
- [AcceptMatch](#)— Per una partita che richiede l'accettazione del giocatore, avvisa Amazon GameLift Servers quando un giocatore accetta una partita proposta.
- [StopMatchmaking](#)— Annulla una richiesta di matchmaking.
- [StartMatchBackfill](#)- Richiedi altre partite tra giocatori per riempire gli slot vuoti in una sessione di gioco esistente.

Linguaggi di programmazione compatibili

L' AWS SDK con supporto per Amazon GameLift Servers è disponibile nelle seguenti lingue. Per informazioni sul supporto per gli ambienti di sviluppo, consulta la documentazione di ogni lingua.

- C++ ([documenti SDK](#)) ([Amazon GameLift Servers](#))
- Java (documenti [SDK](#)) ([Amazon GameLift Servers](#))
- .NET (documenti [SDK](#)) ([Amazon GameLift Servers](#))
- Go (documenti [SDK](#)) ([Amazon GameLift Servers](#))
- Python (documenti [SDK](#)) ([Amazon GameLift Servers](#))
- [Ruby \(documenti SDK\)](#) ([Amazon GameLift Servers](#))
- [PHP \(documenti SDK\)](#) ([Amazon GameLift Servers](#))
- JavaScript/Node.js (documenti [SDK](#)) ([Amazon GameLift Servers](#))

FlexMatch linguaggio delle regole

Gli argomenti di riferimento in questa sezione descrivono la sintassi e la semantica utilizzate per creare regole di matchmaking da utilizzare con Amazon GameLift Servers FlexMatch. Per informazioni dettagliate sulla stesura di regole e set di regole di matchmaking, vedere [Costruisci un FlexMatch set di regole](#).

Argomenti

- [FlexMatch schema del set di regole](#)
- [FlexMatch definizioni delle proprietà del set di regole](#)
- [FlexMatch tipi di regole](#)
- [FlexMatch espressioni di proprietà](#)

FlexMatch schema del set di regole

FlexMatch i set di regole utilizzano uno schema standard per le regole per partite piccole e grandi. Per una descrizione dettagliata di ogni sezione, vedere. [FlexMatch definizioni delle proprietà del set di regole](#)

Schema del set di regole per partite di piccole dimensioni

Lo schema seguente documenta tutte le possibili proprietà e i valori consentiti per un set di regole utilizzato per creare partite con un massimo di 40 giocatori.

```
{
  "name": "string",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "string",
    "type": <"string", "number", "string_list", "string_number_map">,
    "default": "string"
  }],
  "algorithm": {
    "strategy": "exhaustiveSearch",
    "batchingPreference": <"random", "sorted">,
    "sortByAttributes": [ "string" ],
    "expansionAgeSelection": <"newest", "oldest">,
    "backfillPriority": <"normal", "low", "high">
  },
  "teams": [{
    "name": "string",
    "maxPlayers": number,
    "minPlayers": number,
    "quantity": integer
  }],
  "rules": [{
    "type": "distance",
    "name": "string",
    "description": "string",
    "measurements": "string",
    "referenceValue": number,
    "maxDistance": number,
    "minDistance": number,
    "partyAggregation": <"avg", "min", "max">
  }, {
    "type": "comparison",
    "name": "string",
    "description": "string",
    "measurements": "string",
    "referenceValue": number,
    "operation": <"<", "<=", "=", "!=", ">", ">=">,
    "partyAggregation": <"avg", "min", "max">
  }
}
```

```

    },{
      "type": "collection",
      "name": "string",
      "description": "string",
      "measurements": "string",
      "referenceValue": number,
      "operation": <"intersection", "contains", "reference_intersection_count">,
      "maxCount": number,
      "minCount": number,
      "partyAggregation": <"union", "intersection">
    },{
      "type": "latency",
      "name": "string",
      "description": "string",
      "maxLatency": number,
      "maxDistance": number,
      "distanceReference": number,
      "partyAggregation": <"avg", "min", "max">
    },{
      "type": "distanceSort",
      "name": "string",
      "description": "string",
      "sortDirection": <"ascending", "descending">,
      "sortByAttribute": "string",
      "mapKey": <"minValue", "maxValue">,
      "partyAggregation": <"avg", "min", "max">
    },{
      "type": "absoluteSort",
      "name": "string",
      "description": "string",
      "sortDirection": <"ascending", "descending">,
      "sortByAttribute": "string",
      "mapKey": <"minValue", "maxValue">,
      "partyAggregation": <"avg", "min", "max">
    },{
      "type": "compound",
      "name": "string",
      "description": "string",
      "statement": "string"
    }
  ]],
  "expansions": [{
    "target": "string",
    "steps": [{

```

```

        "waitTimeSeconds": number,
        "value": number
    }, {
        "waitTimeSeconds": number,
        "value": number
    }]
}]
}

```

Schema del set di regole per partite di grandi dimensioni

Lo schema seguente documenta tutte le possibili proprietà e i valori consentiti per un set di regole utilizzato per creare partite con più di 40 giocatori. Se il totale dei `maxPlayers` valori per tutte le squadre nel set di regole supera 40, allora FlexMatch elabora le richieste di partite che utilizzano questa regola stabilita nelle linee guida per le partite di grandi dimensioni.

```

{
  "name": "string",
  "ruleLanguageVersion": "1.0",
  "playerAttributes": [{
    "name": "string",
    "type": <"string", "number", "string_list", "string_number_map">,
    "default": "string"
  }],
  "algorithm": {
    "strategy": "balanced",
    "batchingPreference": <"largestPopulation", "fastestRegion">,
    "balancedAttribute": "string",
    "expansionAgeSelection": <"newest", "oldest">,
    "backfillPriority": <"normal", "low", "high">
  },
  "teams": [{
    "name": "string",
    "maxPlayers": number,
    "minPlayers": number,
    "quantity": integer
  }],
  "rules": [{
    "name": "string",
    "type": "latency",
    "description": "string",
    "maxLatency": number,
    "partyAggregation": <"avg", "min", "max">
  }],
}

```

```
    }, {
      "name": "string",
      "type": "batchDistance",
      "batchAttribute": "string",
      "maxDistance": number
    }],
    "expansions": [{
      "target": "string",
      "steps": [{
        "waitTimeSeconds": number,
        "value": number
      }, {
        "waitTimeSeconds": number,
        "value": number
      }]
    }]
  }]
}
```

FlexMatch definizioni delle proprietà del set di regole

Questa sezione definisce ogni proprietà nello schema del set di regole. Per ulteriore assistenza sulla creazione di un set di regole, vedere [Costruisci un FlexMatch set di regole](#).

name

Un'etichetta descrittiva per il set di regole. Questo valore non è associato al nome assegnato al Amazon GameLift Servers [MatchmakingRuleSet risorsa](#). Questo valore è incluso nei dati di matchmaking che descrivono una partita completata, ma non è usato da nessuno Amazon GameLift Servers processi.

Valori consentiti: String

Obbligatorio? No

ruleLanguageVersion

La versione del linguaggio delle espressioni di FlexMatch proprietà in uso.

Valori consentiti: «1.0»

Obbligatorio? Sì

playerAttributes

Una raccolta di dati dei giocatori inclusa nelle richieste di matchmaking e utilizzata nel processo di matchmaking. Qui puoi anche dichiarare gli attributi per includere i dati dei giocatori nei dati di matchmaking che vengono trasmessi ai server di gioco, anche se i dati non vengono utilizzati nel processo di matchmaking.

Obbligatorio? No

name

Un nome univoco per l'attributo del giocatore che deve essere utilizzato dal matchmaker. Questo nome deve corrispondere al nome dell'attributo del giocatore a cui si fa riferimento nelle richieste di matchmaking.

Valori consentiti: String

Obbligatorio? Sì

type

Il tipo di dati del valore dell'attributo del giocatore.

Valori consentiti: «string», «number», «string_list», «string_number_map»

Obbligatorio? Sì

default

Un valore predefinito da utilizzare quando una richiesta di matchmaking non ne fornisce uno per un giocatore.

Valori consentiti: qualsiasi valore consentito per l'attributo player.

Obbligatorio? No

algorithm

Impostazioni di configurazione opzionali per personalizzare il processo di matchmaking.

Obbligatorio? No

strategy

Il metodo da usare per creare partite. Se questa proprietà non è impostata, il comportamento predefinito è «ExhaustiveSearch».

Valori consentiti:

- «ExhaustiveSearch»: metodo di abbinamento standard. FlexMatch crea una corrispondenza basata sul ticket più vecchio di un lotto valutando gli altri ticket del pool in base a una serie di regole di partita personalizzate. Questa strategia viene utilizzata per partite di 40 giocatori o meno. Quando si utilizza questa strategia, `batchingPreference` deve essere impostata su «casuale» o «ordinata».
- «bilanciato»: metodo ottimizzato per formare rapidamente partite di grandi dimensioni. Questa strategia viene utilizzata solo per partite da 41 a 200 giocatori. Crea partite preordinando il pool di biglietti, costruendo potenziali partite e assegnando i giocatori alle squadre, quindi bilanciando ogni squadra in una partita utilizzando un attributo del giocatore specificato. Ad esempio, questa strategia può essere utilizzata per pareggiare i livelli di abilità medi di tutte le squadre in una partita. Quando si utilizza questa strategia, `balancedAttribute` deve essere impostata e `batchingPreference` deve essere impostata su «LargestPopulation» o «FastestRegion». La maggior parte dei tipi di regole personalizzate non viene riconosciuta con questa strategia.

Obbligatorio? Sì

batchingPreference

Il metodo di preordinamento da utilizzare prima di raggruppare i biglietti per il match building. L'ordinamento preliminare del pool di biglietti fa sì che i biglietti vengano raggruppati in base a una caratteristica specifica, che tende ad aumentare l'uniformità tra i giocatori nelle partite finali.

Valori consentiti:

- «random» — Valido solo con `strategy` = «ExhaustiveSearch». Non viene effettuato alcun preordinamento; i biglietti nel pool vengono raggruppati in modo casuale. Questo è il comportamento predefinito per una strategia di ricerca esaustiva.
- «sorted» — Valido solo con `strategy` = «ExhaustiveSearch». Il pool di biglietti è preordinato in base agli attributi del giocatore elencati in `sortByAttributes`
- «LargestPopulation»: valido solo con `strategy` = «balanced». Il pool di biglietti è preordinato per regioni in cui i giocatori segnalano livelli di latenza accettabili. Questo è il comportamento predefinito per una strategia equilibrata.
- «fastestRegion» — Valido solo con `strategy` = «balanced». Il pool di biglietti è preordinato per regioni in cui i giocatori segnalano i livelli di latenza più bassi. Le partite risultanti

richiedono più tempo per essere completate, ma la latenza per tutti i giocatori tende ad essere bassa.

Obbligatorio? Sì

balancedAttribute

Il nome di un attributo del giocatore da usare quando si costruiscono partite di grandi dimensioni con la strategia bilanciata.

Valori consentiti: qualsiasi attributo dichiarato `playerAttributes` con `type = «numero»`.

Obbligatorio? Sì, se `strategy = «bilanciato»`.

sortByAttributes

Un elenco di attributi dei giocatori da utilizzare per preordinare il pool di biglietti prima del raggruppamento in batch. Questa proprietà viene utilizzata solo per il preordinamento con la strategia di ricerca esaustiva. L'ordine dell'elenco degli attributi determina l'ordinamento. FlexMatch utilizza una convenzione di ordinamento standard per i valori alfabetici e numerici.

Valori consentiti: qualsiasi attributo dichiarato in `playerAttributes`

Obbligatorio? Sì, if `batchingPreference = «ordinato»`.

backfillPriority

Il metodo di assegnazione delle priorità per abbinare i ticket di riserva. Questa proprietà determina quando FlexMatch elabora i ticket di backfill in un batch. Viene utilizzato solo per il preordinamento con la strategia di ricerca esaustiva. Se questa proprietà non è impostata, il comportamento predefinito è «normale».

Valori consentiti:

- «normale»: il tipo di richiesta di un ticket (backfill o nuova partita) non viene considerato durante la creazione delle partite.
- «alto»: un batch di ticket viene ordinato per tipo di richiesta (e quindi per età) e FlexMatch tenta innanzitutto di abbinare i ticket backfill.
- «basso»: un batch di ticket viene ordinato per tipo di richiesta (e quindi per età) e FlexMatch tenta innanzitutto di abbinare i ticket non riempiti.

Obbligatorio? No

expansionAgeSelection

Il metodo per calcolare il tempo di attesa per l'espansione di una regola di partita. Le espansioni vengono utilizzate per ridurre i requisiti di una partita se una partita non viene completata dopo un certo periodo di tempo. Il tempo di attesa viene calcolato in base all'età dei biglietti già presenti nella partita parzialmente riempita. Se questa proprietà non è impostata, il comportamento predefinito è «più recente».

Valori consentiti:

- «più recente»: il tempo di attesa per l'espansione viene calcolato in base al ticket con il timestamp di creazione più recente della partita parzialmente completata. Le espansioni tendono ad essere attivate più lentamente, perché un ticket più recente può far ripartire il tempo di attesa.
- «più vecchio»: il tempo di attesa per l'espansione viene calcolato in base al ticket con la data di creazione più vecchia della partita. Le espansioni tendono ad essere attivate più rapidamente.

Obbligatorio? No

teams

La configurazione delle squadre in una partita. Fornisci un nome e un intervallo di dimensioni per ogni squadra. Un set di regole deve definire almeno una squadra.

name

Un nome univoco per il team. È possibile fare riferimento ai nomi dei team nelle regole e nelle espansioni. In caso di partita riuscita, i giocatori vengono assegnati in base al nome della squadra nei dati di matchmaking.

Valori consentiti: String

Obbligatorio? Sì

maxPlayers

Il numero massimo di giocatori che possono essere assegnati alla squadra.

Valori consentiti: Numero

Obbligatorio? Sì

minPlayers

Il numero minimo di giocatori che devono essere assegnati alla squadra prima che la partita sia valida.

Valori consentiti: Numero

Obbligatorio? Sì

quantity

Il numero di squadre di questo tipo da creare in una partita. Le squadre con quantità superiori a 1 sono designate con un numero aggiunto («Red_1», «Red_2», ecc.). Se questa proprietà non è impostata, il valore predefinito è «1».

Valori consentiti: Numero

Obbligatorio? No

rules

Una raccolta di regole che definiscono come valutare i giocatori per una partita.

Obbligatorio? No

name

Un nome univoco per la regola. Tutte le regole di un set di regole devono avere nomi univoci. I nomi delle regole vengono citati nei registri degli eventi e nelle metriche che tengono traccia delle attività correlate alla regola.

Valori consentiti: String

Obbligatorio? Sì

description

Una descrizione testuale della regola. Queste informazioni possono essere utilizzate per identificare lo scopo di una regola. Non viene utilizzato nel processo di matchmaking.

Valori consentiti: String

Obbligatorio? No

type

Il tipo di dichiarazione della regola. Ogni tipo di regola ha proprietà aggiuntive che devono essere impostate. Per ulteriori dettagli sulla struttura e l'uso di ogni tipo di regola, vedere [FlexMatch tipi di regole](#).

Valori consentiti:

- «AbsoluteSort»: esegue l'ordinamento utilizzando un metodo di ordinamento esplicito che ordina i ticket in batch in base alla differenza tra uno specifico attributo del giocatore e il ticket più vecchio del batch.
- «collezione»: valuta i valori di una raccolta, ad esempio un attributo del giocatore che è una raccolta o un insieme di valori per più giocatori.
- «confronto»: confronta due valori.
- «composto»: definisce una regola di matchmaking composta utilizzando una combinazione logica di altre regole nel set di regole. Supportato solo per partite con 40 giocatori o meno.
- «distanza»: misura la distanza tra i valori numerici.
- «batchDistance»: misura la differenza tra il valore di un attributo e la utilizza per raggruppare le richieste di corrispondenza.
- «DistanceSort»: esegue l'ordinamento utilizzando un metodo di ordinamento esplicito che ordina i ticket in batch in base alla differenza tra uno specifico attributo del giocatore con un valore numerico e il ticket più vecchio del batch.
- «latenza»: valuta i dati di latenza regionale riportati per una richiesta di matchmaking.

Obbligatorio? Sì

expansions

Regole per allentare i requisiti delle partite nel tempo quando una partita non può essere completata. Imposta le espansioni come una serie di passaggi da applicare gradualmente per facilitare la ricerca degli abbinamenti. Per impostazione predefinita, FlexMatch calcola il tempo di attesa in base all'età del nuovo biglietto aggiunto a una partita. È possibile modificare il modo in cui vengono calcolati i tempi di attesa per l'espansione utilizzando la proprietà `algorithm`.

`expansionAgeSelection`

I tempi di attesa per l'espansione sono valori assoluti, quindi ogni passaggio dovrebbe avere un tempo di attesa più lungo rispetto al passaggio precedente. Ad esempio, per pianificare una serie graduale di espansione, è possibile utilizzare tempi di attesa di 30 secondi, 40 secondi e 50

secondi. I tempi di attesa non possono superare il tempo massimo consentito per una richiesta di abbinamento, impostato nella configurazione del matchmaking.

Obbligatorio? No

target

L'elemento del set di regole da rilassare. È possibile modificare le proprietà relative alle dimensioni del team o qualsiasi proprietà delle istruzioni delle regole. La sintassi è "<component name>[<rule/team name>]. <property name>». Ad esempio, per modificare le dimensioni minime del team: `teams[Red, Yellow].minPlayers`. Per modificare il requisito minimo di abilità in una dichiarazione della regola di confronto denominata «minSkill»: `rules[minSkill].referenceValue`.

Obbligatorio? Sì

steps

waitTimeSeconds

Il periodo di attesa, in secondi, prima di applicare il nuovo valore per l'elemento del set di regole di destinazione.

Obbligatorio? Sì

value

Il nuovo valore per l'elemento del set di regole di destinazione.

FlexMatch tipi di regole

Regola della distanza in batch

```
batchDistance
```

Le regole di distanza in batch misurano la differenza tra i valori di due attributi. È possibile utilizzare il tipo di regola della distanza in batch con corrispondenze grandi e piccole. Esistono due tipi di regole sulla distanza in batch:

- Confronta i valori degli attributi numerici. Ad esempio, una regola di distanza in batch di questo tipo potrebbe richiedere che tutti i giocatori di una partita si trovino a meno di due livelli di abilità l'uno dall'altro. Per questo tipo, definisci una distanza massima tra `batchAttribute` tutti i ticket.

- Confronta i valori degli attributi di stringa. Ad esempio, una regola di distanza in batch di questo tipo potrebbe richiedere che tutti i giocatori di una partita richiedano la stessa modalità di gioco. Per questo tipo, definisci un `batchAttribute` valore che FlexMatch utilizza per formare batch.

Proprietà della regola della distanza in batch

- **batchAttribute**— Il valore dell'attributo player utilizzato per formare i batch.
- **maxDistance**— Il valore della distanza massima per una partita riuscita. Utilizzato per confrontare gli attributi numerici.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono i valori minimo (`minmax`), massimo (`max`) e medio (`avg`) per i giocatori di un ticket. Il valore predefinito è `avg`.

Example

Examples (Esempi)

```
{
  "name": "SimilarSkillRatings",
  "description": "All players must have similar skill ratings",
  "type": "batchDistance",
  "batchAttribute": "SkillRating",
  "maxDistance": "500"
}
```

```
{
  "name": "SameGameMode",
  "description": "All players must have the same game mode",
  "type": "batchDistance",
  "batchAttribute": "GameMode"
}
```

Regola di confronto

```
comparison
```

Le regole di confronto confrontano il valore di un attributo del giocatore con un altro valore. Esistono due tipi di regole di confronto:

- **Confronta con il valore di riferimento.** Ad esempio, una regola di confronto di questo tipo potrebbe richiedere che i giocatori abbinati abbiano un certo livello di abilità o superiore. Per questo tipo, specifica un attributo del giocatore, un valore di riferimento e un'operazione di confronto.
- **Confronta i giocatori.** Ad esempio, una regola di confronto di questo tipo potrebbe richiedere che tutti i giocatori della partita utilizzino personaggi diversi. Per questo tipo, specifica un attributo del giocatore e l'operazione di confronto uguale (=) o non uguale (!=). Non specificate un valore di riferimento.

Note

Le regole della distanza in batch sono più efficienti per confrontare gli attributi dei giocatori. Per ridurre la latenza del matchmaking, usa una regola di distanza in batch quando possibile.

Proprietà della regola di confronto

- **measurements**— Il valore dell'attributo del giocatore da confrontare.
- **referenceValue**— Il valore con cui confrontare la misurazione per una potenziale partita.
- **operation**— Il valore che determina come confrontare la misurazione con il valore di riferimento. Le operazioni valide includono: <<=,=,,!=,>,>=.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono i valori minimo (minmax), massimo () e medio (avg) per i giocatori di un ticket. Il valore predefinito è avg.

Regola della distanza

distance

Le regole sulla distanza misurano la differenza tra due valori numerici, ad esempio la distanza tra i livelli di abilità dei giocatori. Ad esempio, una regola sulla distanza potrebbe richiedere che tutti i giocatori abbiano giocato una partita per almeno 30 ore.

Note

Le regole della distanza in batch sono più efficienti per confrontare gli attributi dei giocatori. Per ridurre la latenza del matchmaking, usa una regola di distanza in batch quando possibile.

Proprietà della regola di distanza

- **measurements**— Il valore dell'attributo del giocatore per cui misurare la distanza. Deve essere un attributo con un valore numerico.
- **referenceValue**— Il valore numerico rispetto al quale misurare la distanza per una potenziale corrispondenza.
- **minDistance/maxDistance**— Il valore della distanza minima o massima per una partita riuscita.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono i valori minimo (minmax), massimo () e medio (avg) per i giocatori di un ticket. Il valore predefinito è avg.

Regola di raccolta

collection

Le regole di raccolta confrontano i valori degli attributi di un gruppo di giocatori con quelli degli altri giocatori del gruppo o con un valore di riferimento. Una raccolta può contenere valori di attributi per più giocatori, un attributo del giocatore come elenco di stringhe o entrambi. Ad esempio, una regola di raccolta potrebbe riguardare i personaggi scelti dai giocatori di una squadra. La regola potrebbe quindi richiedere che la squadra abbia almeno uno di un determinato personaggio.

Proprietà della regola di raccolta

- **measurements**— La raccolta di valori degli attributi dei giocatori da confrontare. I valori degli attributi devono essere elenchi di stringhe.
- **referenceValue**— Il valore (o la raccolta di valori) da utilizzare per confrontare le misurazioni per una potenziale corrispondenza.
- **operation**— Il valore che determina come confrontare una raccolta di misurazioni. Le operazioni valide includono quanto segue:

- **intersection**— Questa operazione misura il numero di valori uguali nelle collezioni di tutti i giocatori. Per un esempio di regola che utilizza l'operazione di intersezione, vedi. [Esempio: utilizza l'ordinamento esplicito per trovare le migliori corrispondenze](#)
- **contains**— Questa operazione misura il numero di raccolte di attributi dei giocatori che contengono il valore di riferimento specificato. Per un esempio di regola che utilizza l'operazione contains, vedere [Esempio: imposta i requisiti a livello di team e i limiti di latenza](#).
- **reference_intersection_count**— Questa operazione misura il numero di elementi in una raccolta di attributi del giocatore che corrispondono agli elementi nella raccolta di valori di riferimento. È possibile utilizzare questa operazione per confrontare più attributi diversi dei giocatori. Per un esempio di regola che confronta le raccolte di attributi di più giocatori, vedi [Esempio: trova le intersezioni tra gli attributi di più giocatori](#).
- **minCount/maxCount**— Il valore di conteggio minimo o massimo per una partita riuscita.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Per questo valore, puoi usarlo `union` per combinare gli attributi dei giocatori di tutti i giocatori del gruppo. In alternativa, puoi `intersection` usare gli attributi dei giocatori che il gruppo ha in comune. Il valore predefinito è `union`.

Regola composta

compound

Le regole composte utilizzano istruzioni logiche per formare partite di 40 giocatori o meno. È possibile utilizzare più regole composte in un unico set di regole. Quando si utilizzano più regole composte, tutte le regole composte devono essere vere per formare una corrispondenza.

Non è possibile espandere una regola composta utilizzando [regole di espansione](#), ma è possibile espandere le regole sottostanti o di supporto.

Proprietà delle regole composte

- **statement**— La logica utilizzata per combinare singole regole per formare la regola composta. Le regole specificate in questa proprietà devono essere state definite in precedenza nel set di regole. Non è possibile utilizzare `batchDistance` regole in una regola composta.

Questa proprietà supporta i seguenti operatori logici:

- **and**— L'espressione è vera se i due argomenti forniti sono veri.

- **or**— L'espressione è vera se uno dei due argomenti forniti è vero.
- **not**— Inverte il risultato dell'argomento nell'espressione.
- **xor**— L'espressione è vera se solo uno degli argomenti è vero.

Example Esempio

L'esempio seguente abbina giocatori con diversi livelli di abilità in base alla modalità di gioco selezionata.

```
{
  "name": "CompoundRuleExample",
  "type": "compound",
  "statement": "or(and(SeriousPlayers, VeryCloseSkill), and(CasualPlayers, SomewhatCloseSkill))"
}
```

Latency rule (Regola di latenza)

latency

Le regole di latenza misurano la latenza dei giocatori per località. Una regola di latenza ignora qualsiasi posizione con una latenza superiore al massimo. Un giocatore deve avere un valore di latenza inferiore al massimo in almeno una località affinché la regola di latenza lo accetti. È possibile utilizzare questo tipo di regola con corrispondenze di grandi dimensioni specificando la proprietà `maxLatency`.

Proprietà della regola di latenza

- **maxLatency**— Il valore di latenza massimo accettabile per una posizione. Se un ticket non ha sedi con una latenza inferiore al massimo, il ticket non corrisponde alla regola di latenza.
- **maxDistance**— Il valore massimo tra la latenza di ogni ticket e il valore di riferimento della distanza.
- **distanceReference**— Il valore di latenza con cui confrontare la latenza del ticket. I ticket che rientrano nella distanza massima del valore di riferimento della distanza determinano una partita vincente. Le opzioni valide includono i valori di latenza minima (`minavg`) e media (`()`) del giocatore.

- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono i valori minimo (minmax), massimo () e medio (avg) per i giocatori di un ticket. Il valore predefinito è avg.

Note

Una coda può collocare una sessione di gioco in una regione che non corrisponde a una regola di latenza. Per ulteriori informazioni sui criteri di latenza per le code, consulta [Creare un criterio di latenza dei giocatori](#).

Regola di ordinamento assoluto

```
absoluteSort
```

Le regole di ordinamento assoluto ordinano un batch di ticket di matchmaking in base a un attributo del giocatore specificato rispetto al primo ticket aggiunto al batch.

Proprietà della regola di ordinamento assoluto

- **sortDirection**— L'ordine in cui ordinare i ticket di matchmaking. Le opzioni valide includono ascending e descending
- **sortAttribute**— L'attributo del giocatore in base al quale ordinare i ticket.
- **mapKey**— Le opzioni per ordinare l'attributo del giocatore se si tratta di una mappa. Le opzioni valide includono:
 - minvalue— La chiave con il valore più basso è la prima.
 - maxvalue— La chiave con il valore più alto viene prima.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono l'attributo giocatore minimo (min), l'attributo giocatore massimo (max) e la media (avg) di tutti gli attributi del giocatore per i giocatori del gruppo. Il valore predefinito è avg.

Example

Esempio

La seguente regola di esempio ordina i giocatori in base al livello di abilità e calcola la media del livello di abilità dei gruppi.

```
{
  "name": "AbsoluteSortExample",
  "type": "absoluteSort",
  "sortDirection": "ascending",
  "sortAttribute": "skill",
  "partyAggregation": "avg"
}
```

Regola di ordinamento a distanza

```
distanceSort
```

Le regole di ordinamento a distanza ordinano un batch di ticket di matchmaking in base alla distanza di un attributo del giocatore specificato dal primo ticket aggiunto al batch.

Proprietà della regola di ordinamento della distanza

- **sortDirection**— La direzione per ordinare i ticket di matchmaking. Le opzioni valide includono `ascending` e `descending`
- **sortAttribute**— L'attributo del giocatore in base al quale ordinare i ticket.
- **mapKey**— Le opzioni per ordinare l'attributo del giocatore se si tratta di una mappa. Le opzioni valide includono:
 - `minValue`— Per il primo ticket aggiunto al batch, trova la chiave con il valore più basso.
 - `maxValue`— Per il primo biglietto aggiunto al lotto, trova la chiave con il valore più alto.
- **partyAggregation**— Il valore che determina come FlexMatch gestisce i ticket con più giocatori (party). Le opzioni valide includono i valori minimo (`minmax`), massimo (`max`) e medio (`avg`) per i giocatori di un ticket. Il valore predefinito è `avg`.

FlexMatch espressioni di proprietà

Le espressioni di proprietà possono essere utilizzate per definire determinate proprietà relative al matchmaking. Consentono di utilizzare calcoli e logica per definire il valore di una proprietà. Le espressioni di proprietà si presentano generalmente in una delle due forme seguenti:

- Dati dei singoli giocatori.

- Raccolte calcolate di dati dei singoli giocatori.

Espressioni di proprietà di matchmaking comuni

Un'espressione di proprietà identifica un valore specifico per un giocatore, una squadra o una partita. Le seguenti espressioni parziali illustrano come identificare team e giocatori:

Obiettivo	Input	Significato	Output
Per identificare un determinato team in un abbinamento:	<code>teams[red]</code>	Team "red"	Team
Per identificare un insieme di squadre specifiche in una partita:	<code>teams[red,blue]</code>	La squadra rossa e la squadra blu	List<Team>
Per identificare tutti i team in un abbinamento:	<code>teams[*]</code>	Tutti i team	List<Team>
Per identificare i giocatori in un determinato team:	<code>team[red].players</code>	Giocatori del team "red"	List<Player>
Per identificare i giocatori in una serie di squadre specifiche in una partita:	<code>team[red,blue].players</code>	Giocatori nell'abbinamento, raggruppati per team	List<List<Player>>
Per identificare i giocatori in un abbinamento:	<code>team[*].players</code>	Giocatori nell'abbinamento, raggruppati per team	List<List<Player>>

Esempi di espressioni di proprietà

La tabella seguente illustra alcune espressioni di proprietà basate sugli esempi precedenti:

Expression	Significato	Tipo risultante
<code>teams[red].players[playerId]</code>	Il giocatore IDs di tutti i giocatori della squadra rossa	List<string>
<code>teams[red].players.attributes[skill]</code>	Attributi "skill" di tutti i giocatori del team "red"	List<number>
<code>teams[red,blue].players.attributes[skill]</code>	Gli attributi di «abilità» di tutti i giocatori della squadra rossa e della squadra blu, raggruppati per squadra	List<List<number>>
<code>teams[*].players.attributes[skill]</code>	Attributi "skill" di tutti i giocatori dell'abbinamento, raggruppati per team	List<List<number>>

Aggregazioni di espressioni di proprietà

Le espressioni di proprietà possono essere utilizzate per aggregare i dati dei team utilizzando le seguenti funzioni o combinazioni di funzioni:

Aggregazione	Input	Significato	Output
min	List<number>	Ottiene il minimo di tutti i numeri nell'elenco.	number
max	List<number>	Ottiene il massimo di tutti i numeri nell'elenco.	number
avg	List<number>	Ottiene la media di tutti i numeri nell'elenco.	number

Aggregazione	Input	Significato	Output
median	List<number>	Ottiene la mediana di tutti i numeri nell'elenco.	number
sum	List<number>	Ottiene la somma di tutti i numeri nell'elenco.	number
count	List<?>	Ottiene il numero di elementi nell'elenco.	number
stddev	List<number>	Ottiene la deviazione e standard di tutti i numeri nell'elenco.	number
flatten	List<List<?>>	Trasforma una raccolta di elenchi nidificati in un singolo elenco contenente tutti gli elementi.	List<?>
set_intersection	List<List<string>>	Ottiene un elenco di stringhe che si trovano in tutti gli elenchi di stringhe di una raccolta.	List<string>
Tutti i formati sopra elencati	List<List<?>>	Tutte le operazioni su un elenco nidificato o operano singolarmente su ogni sottoelenco per ottenere un elenco di risultati.	List<?>

La tabella riportata di seguito mostra alcune espressioni di proprietà valide che utilizzano le funzioni di aggregazione:

Expression	Significato	Tipo risultante
<code>flatten(teams[*].players.attributes[skill])</code>	Attributi "skill" di tutti i giocatori dell'abbinamento (non raggruppati)	List<number>
<code>avg(teams[red].players.attributes[skill])</code>	Livello di competenza a medio dei giocatori del team "red"	number
<code>avg (teams [*] .players.attributes [abilità])</code>	Livello di competenza a medio di ogni team nell'abbinamento	List<number>
<code>avg(flatten(teams[*].players.attributes[skill]))</code>	Livello di competenza a medio di tutti i giocatori dell'abbinamento. L'espressione ottiene un elenco appiattito delle competenze dei giocatori e quindi ne calcola la media.	number
<code>count(teams[red].players)</code>	Numero di giocatori del team "red"	number
<code>count (teams[*].players)</code>	Numero di giocatori di ogni team nell'abbinamento	List<number>
<code>max(avg(teams[*].players.attributes[skill]))</code>	Livello di competenza a del team più elevato nella corrispondenza	number

FlexMatch eventi di matchmaking

Amazon GameLift Servers FlexMatch emette eventi per ogni ticket di matchmaking man mano che viene elaborato. Puoi pubblicare questi eventi su un argomento di Amazon SNS, come descritto in [Configurazione FlexMatch notifiche di eventi](#). Questi eventi vengono inoltre trasmessi ad Amazon CloudWatch Events quasi in tempo reale e con la massima diligenza possibile.

Questo argomento descrive la struttura di FlexMatch eventi e fornisce un esempio per ogni tipo di evento. Per ulteriori informazioni sullo stato dei ticket di matchmaking, vedi [MatchmakingTicket](#) nel Amazon GameLift Servers Riferimento API.

Argomenti

- [MatchmakingSearching](#)
- [PotentialMatchCreated](#)
- [AcceptMatch](#)
- [AcceptMatchCompleted](#)
- [MatchmakingSucceeded](#)
- [MatchmakingTimedOut](#)
- [MatchmakingCancelled](#)
- [MatchmakingFailed](#)

MatchmakingSearching

Il ticket è stato inserito nell'abbinamento. Questo include le nuove richieste e le richieste che facevano parte di un abbinamento proposto non riuscito.

Risorsa: ConfigurationArn

Dettagli: tipo, biglietti estimatedWaitMillis, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "cc3d3ebe-1d90-48f8-b268-c96655b8f013",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
```

```
"time": "2017-08-08T21:15:36.421Z",
"region": "us-west-2",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
],
"detail": {
  "tickets": [
    {
      "ticketId": "ticket-1",
      "startTime": "2017-08-08T21:15:35.676Z",
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  ],
  "estimatedWaitMillis": "NOT_AVAILABLE",
  "type": "MatchmakingSearching",
  "gameSessionInfo": {
    "players": [
      {
        "playerId": "player-1"
      }
    ]
  }
}
}
```

PotentialMatchCreated

Un potenziale abbinamento è stato creato. Questo ticket viene emesso per tutti i nuovi abbinamenti potenziali, che l'accettazione sia obbligatoria o meno.

Risorsa: ConfigurationArn

Dettagli: tipo, ticket, AcceptanceTimeout, AcceptanceRequired,, MatchID ruleEvaluationMetrics
gameSessionInfo

Esempio

```
{
```

```
"version": "0",
"id": "fce8633f-aea3-45bc-aeba-99d639cad2d4",
"detail-type": "GameLift Matchmaking Event",
"source": "aws.gamelift",
"account": "123456789012",
"time": "2017-08-08T21:17:41.178Z",
"region": "us-west-2",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
],
"detail": {
  "tickets": [
    {
      "ticketId": "ticket-1",
      "startTime": "2017-08-08T21:15:35.676Z",
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    },
    {
      "ticketId": "ticket-2",
      "startTime": "2017-08-08T21:17:40.657Z",
      "players": [
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    }
  ],
  "acceptanceTimeout": 600,
  "ruleEvaluationMetrics": [
    {
      "ruleName": "EvenSkill",
      "passedCount": 3,
      "failedCount": 0
    },
    {
      "ruleName": "EvenTeams",
      "passedCount": 3,
```

```
    "failedCount": 0
  },
  {
    "ruleName": "FastConnection",
    "passedCount": 3,
    "failedCount": 0
  },
  {
    "ruleName": "NoobSegregation",
    "passedCount": 3,
    "failedCount": 0
  }
],
"acceptanceRequired": true,
"type": "PotentialMatchCreated",
"gameSessionInfo": {
  "players": [
    {
      "playerId": "player-1",
      "team": "red"
    },
    {
      "playerId": "player-2",
      "team": "blue"
    }
  ]
},
"matchId": "3faf26ac-f06e-43e5-8d86-08feff26f692"
}
```

AcceptMatch

I giocatori hanno accettato un potenziale abbinamento. Questo evento contiene l'attuale stato di accettazione di ogni giocatore dell'abbinamento. I dati mancanti indicano che non è stato chiamato per quel giocatore. AcceptMatch

Risorsa: ConfigurationArn

Dettagli: tipo, biglietti, MatchID, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "b3f76d66-c8e5-416a-aa4c-aa1278153edc",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:04:42.660Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T20:04:16.637Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue",
            "accepted": false
          }
        ]
      }
    ]
  },
  "type": "AcceptMatch",
  "gameSessionInfo": {
    "players": [
      {
        "playerId": "player-1",
```

```
        "team": "red"
      },
      {
        "playerId": "player-2",
        "team": "blue",
        "accepted": false
      }
    ]
  },
  "matchId": "848b5f1f-0460-488e-8631-2960934d13e5"
}
```

AcceptMatchCompleted

L'accettazione dell'abbinamento è completata a seguito dell'accettazione di un giocatore, del rifiuto di un giocatore o del timeout dell'accettazione.

Risorsa: ConfigurationArn

Dettagli: tipo, biglietti, accettazione, matchID, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "b1990d3d-f737-4d6c-b150-af5ace8c35d3",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T20:43:14.621Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T20:30:40.972Z",
        "players": [
```

```
    {
      "playerId": "player-1",
      "team": "red"
    }
  ],
},
{
  "ticketId": "ticket-2",
  "startTime": "2017-08-08T20:33:14.111Z",
  "players": [
    {
      "playerId": "player-2",
      "team": "blue"
    }
  ]
}
],
"acceptance": "TimedOut",
"type": "AcceptMatchCompleted",
"gameSessionInfo": {
  "players": [
    {
      "playerId": "player-1",
      "team": "red"
    },
    {
      "playerId": "player-2",
      "team": "blue"
    }
  ]
},
"matchId": "a0d9bd24-4695-4f12-876f-ea6386dd6dce"
}
}
```

MatchmakingSucceeded

L'abbinamento è stato completato ed è stata creata una sessione di gioco.

Risorsa: ConfigurationArn

Dettaglio: tipo, biglietti, matchID, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "5ccb6523-0566-412d-b63c-1569e00d023d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T19:59:09.159Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:58:59.277Z",
        "players": [
          {
            "playerId": "player-1",
            "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T19:59:08.663Z",
        "players": [
          {
            "playerId": "player-2",
            "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
            "team": "blue"
          }
        ]
      }
    ]
  },
  "type": "MatchmakingSucceeded",
  "gameSessionInfo": {
    "gameSessionArn": "arn:aws:gamelift:us-west-2:123456789012:gamesession/836cf48d-
bcb0-4a2c-bec1-9c456541352a",

```



```
    "ipAddress": "192.168.1.1",
    "port": 10777,
    "players": [
      {
        "playerId": "player-1",
        "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
        "team": "red"
      },
      {
        "playerId": "player-2",
        "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
        "team": "blue"
      }
    ]
  },
  "matchId": "c0ec1a54-7fec-4b55-8583-76d67adb7754"
}
```

MatchmakingTimedOut

Il ticket di abbinamento non è andato a buon fine perché scaduto.

Risorsa: ConfigurationArn

Dettaglio: tipo, ticket, messaggio ruleEvaluationMetrics, matchID, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "fe528a7d-46ad-4bdc-96cb-b094b5f6bf56",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:11:35.598Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
  ],
  "detail": {
    "reason": "TimedOut",
```

```
"tickets": [
  {
    "ticketId": "ticket-1",
    "startTime": "2017-08-09T20:01:35.305Z",
    "players": [
      {
        "playerId": "player-1",
        "team": "red"
      }
    ]
  }
],
"ruleEvaluationMetrics": [
  {
    "ruleName": "EvenSkill",
    "passedCount": 3,
    "failedCount": 0
  },
  {
    "ruleName": "EvenTeams",
    "passedCount": 3,
    "failedCount": 0
  },
  {
    "ruleName": "FastConnection",
    "passedCount": 3,
    "failedCount": 0
  },
  {
    "ruleName": "NoobSegregation",
    "passedCount": 3,
    "failedCount": 0
  }
],
"type": "MatchmakingTimedOut",
"message": "Removed from matchmaking due to timing out.",
"gameSessionInfo": {
  "players": [
    {
      "playerId": "player-1",
      "team": "red"
    }
  ]
}
}
```

```
}  
}
```

MatchmakingCancelled

Il ticket di abbinamento è stato annullato.

Risorsa: ConfigurationArn

Dettaglio: tipo, ticket, messaggio ruleEvaluationMetrics, matchID, gameSessionInfo

Esempio

```
{  
  "version": "0",  
  "id": "8d6f84da-5e15-4741-8d5c-5ac99091c27f",  
  "detail-type": "GameLift Matchmaking Event",  
  "source": "aws.gamelift",  
  "account": "123456789012",  
  "time": "2017-08-09T20:00:07.843Z",  
  "region": "us-west-2",  
  "resources": [  
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/  
SampleConfiguration"  
  ],  
  "detail": {  
    "reason": "Cancelled",  
    "tickets": [  
      {  
        "ticketId": "ticket-1",  
        "startTime": "2017-08-09T19:59:26.118Z",  
        "players": [  
          {  
            "playerId": "player-1"  
          }  
        ]  
      }  
    ]  
  },  
  "ruleEvaluationMetrics": [  
    {  
      "ruleName": "EvenSkill",  
      "passedCount": 0,  
      "failedCount": 0  
    }  
  ]  
}
```

```
    },
    {
      "ruleName": "EvenTeams",
      "passedCount": 0,
      "failedCount": 0
    },
    {
      "ruleName": "FastConnection",
      "passedCount": 0,
      "failedCount": 0
    },
    {
      "ruleName": "NoobSegregation",
      "passedCount": 0,
      "failedCount": 0
    }
  ],
  "type": "MatchmakingCancelled",
  "message": "Cancelled by request.",
  "gameSessionInfo": {
    "players": [
      {
        "playerId": "player-1"
      }
    ]
  }
}
```

MatchmakingFailed

Il ticket di abbinamento ha riscontrato un errore. Questo potrebbe essere dovuto al fatto che la coda della sessione di gioco non è accessibile o a un errore interno.

Risorsa: ConfigurationArn

Dettaglio: tipo, ticket, messaggio ruleEvaluationMetrics, matchID, gameSessionInfo

Esempio

```
{
  "version": "0",
  "id": "025b55a4-41ac-4cf4-89d1-f2b3c6fd8f9d",
```

```
"detail-type": "GameLift Matchmaking Event",
"source": "aws.gamelift",
"account": "123456789012",
"time": "2017-08-16T18:41:09.970Z",
"region": "us-west-2",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/
SampleConfiguration"
],
"detail": {
  "tickets": [
    {
      "ticketId": "ticket-1",
      "startTime": "2017-08-16T18:41:02.631Z",
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    }
  ],
  "customEventData": "foo",
  "type": "MatchmakingFailed",
  "reason": "UNEXPECTED_ERROR",
  "message": "An unexpected error was encountered during match placing.",
  "gameSessionInfo": {
    "players": [
      {
        "playerId": "player-1",
        "team": "red"
      }
    ]
  },
  "matchId": "3ea83c13-218b-43a3-936e-135cc570cba7"
}
```

Amazon GameLift Servers note di rilascio e versioni SDK

Il Amazon GameLift Servers le note di rilascio forniscono dettagli sulle novità Amazon GameLift Servers funzionalità, aggiornamenti e correzioni relativi al servizio, tra cui FlexMatch funzionalità. Puoi anche trovare il Amazon GameLift Servers cronologia delle versioni per tutti SDKs e per i plugin.

- [Amazon GameLift Servers versioni SDK](#)
- [Amazon GameLift Servers note di rilascio](#)

Amazon GameLift Servers risorse per sviluppatori

Per visualizzare tutto Amazon GameLift Servers documentazione e risorse per sviluppatori, consulta il [Amazon GameLift Servers](#) home page della documentazione.

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.