



FlexMatch Guida per sviluppatori

# Amazon GameLift



Version

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

# Amazon GameLift: FlexMatch Guida per sviluppatori

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

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

---

# Table of Contents

Che cos'è Amazon GameLift FlexMatch? .....	1
FlexMatch Caratteristiche principali .....	2
FlexMatch con GameLift hosting Amazon .....	3
Prezzi per Amazon GameLift FlexMatch .....	3
Funzionamento di FlexMatch .....	4
Componenti del matchmaking .....	4
FlexMatch processo di matchmaking .....	5
Regioni AWS supportate .....	8
Configurazione .....	9
Nozioni di base .....	11
Integrazione per il matchmaking autonomo .....	11
Integrazione con Amazon GameLift hosting .....	13
Costruire un FlexMatch matchmaker .....	15
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
Crea set di regole .....	31
Esempi di set di regole .....	34
Crea una configurazione di matchmaking .....	59
Crea un matchmaker per l'hosting Amazon GameLift .....	59
Crea un matchmaker indipendente FlexMatch .....	62
Modifica una configurazione di matchmaking .....	64
Configurare le notifiche degli eventi .....	64
Configurare EventBridge eventi .....	65
Configura un argomento Amazon SNS .....	65
Configura un argomento SNS con crittografia lato server .....	67
Configurare una sottoscrizione agli argomenti per richiamare una funzione Lambda .....	68
Preparazione di giochi per FlexMatch .....	70
Aggiungi FlexMatch a un client di gioco .....	70
Preparati a richiedere il matchmaking per i giocatori .....	71
Richiedi il matchmaking per i giocatori .....	72

Tieni traccia degli eventi di matchmaking .....	74
Richiedi l'accettazione del giocatore .....	74
Connettiti a una partita .....	75
Esempi di richieste di matchmaking .....	76
Aggiungi FlexMatch a un server GameLift di gioco ospitato su Amazon .....	77
Configura il tuo server di gioco per il matchmaking .....	78
Lavora con i dati del matchmaker .....	79
Riempi i giochi esistenti .....	80
Attiva il riempimento automatico .....	81
Invia richieste di backfill (da un server di gioco) .....	82
Inviare richieste di riempimento (da un servizio clienti) .....	84
Aggiorna i dati delle partite sul server di gioco .....	88
Riferimento FlexMatch .....	89
FlexMatchRiferimento API (AWSSDK) .....	89
Imposta regole e processi di matchmaking .....	89
Richiedi una partita per uno o più giocatori .....	90
Linguaggi di programmazione compatibili .....	90
Lingua delle regole .....	91
Schema del set di regole .....	91
Definizioni delle proprietà dei set di regole .....	95
Tipi di regole .....	102
espressioni di proprietà .....	109
Eventi di matchmaking .....	113
MatchmakingSearching .....	114
PotentialMatchCreated .....	115
AcceptMatch .....	117
AcceptMatchCompleted .....	118
MatchmakingSucceeded .....	120
MatchmakingTimedOut .....	121
MatchmakingCancelled .....	123
MatchmakingFailed .....	125
Protezione con FlexMatch .....	127
Note di rilascio e versioni SDK .....	128
Tutte le GameLift guide Amazon .....	129
Glossario per AWS .....	130
.....	cxxx

# Che cos'è Amazon GameLift FlexMatch?

Amazon GameLift FlexMatch è un servizio di matchmaking personalizzabile per giochi multiplayer. Con FlexMatch, puoi creare un set di regole personalizzato che definisce l'aspetto di una partita multiplayer per il tuo gioco e determina 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.

FlexMatch Utilizzalo come servizio di matchmaking autonomo o integrato con una soluzione di hosting di GameLift giochi Amazon. Ad esempio, potresti implementarla FlexMatch come funzionalità autonoma con giochi con un' peer-to-peer architettura o giochi che utilizzano altre soluzioni di cloud computing. Oppure puoi aggiungere FlexMatch al tuo hosting EC2 GameLift gestito da Amazon o hosting locale con Amazon. GameLift Anywhere Questa guida fornisce informazioni dettagliate su come creare un sistema di FlexMatch 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?

## Sei pronto per iniziare a lavorare con? FlexMatch

Per step-by-step indicazioni su come avviare e utilizzare il gioco FlexMatch, consulta i seguenti argomenti:

- [FlexMatchintegrazione con Amazon GameLift hosting](#)
- [GameLiftFlexMatchIntegrazione con Amazon per il matchmaking autonomo](#)

# FlexMatch Caratteristiche principali

Le seguenti funzionalità sono disponibili in tutti gli FlexMatch scenari, indipendentemente dal fatto che vengano utilizzate FlexMatch come servizio autonomo o con l'hosting di GameLift giochi Amazon.

- **Abbinamento personalizzabile tra 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 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 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 nel pool di matchmaking.

Supporto per i gruppi di giocatori. Genera partite per gruppi di giocatori che vogliono giocare insieme nella stessa squadra. FlexMatch Utilizzalo per trovare altri giocatori per completare la partita secondo necessità.

- **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 GameLift hosting Amazon

FlexMatch offre le seguenti funzionalità aggiuntive da utilizzare con i giochi ospitati con Amazon GameLift. Ciò include giochi con server di gioco personalizzati o server in tempo reale.

- **Posizionamento della sessione di gioco.** Quando una partita viene effettuata con successo, richiede FlexMatch automaticamente ad Amazon un nuovo posizionamento nella sessione di gioco GameLift. I dati generati durante il matchmaking, inclusi gli ID dei giocatori e le assegnazioni delle 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, il posizionamento delle sessioni di gioco con Amazon GameLift può anche utilizzare dati regionali sulla latenza dei giocatori, se forniti.
- **Riempimento automatico delle partite.** Con questa funzionalità abilitata, invia FlexMatch automaticamente una richiesta di rientro 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 FleetiQ con giochi ospitati con risorse Amazon Elastic Compute Cloud (Amazon EC2) Elastic Compute Cloud (Amazon EC2), implementalo come servizio autonomo.

FlexMatch

## Prezzi per Amazon GameLift FlexMatch

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

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

# Come GameLift FlexMatch funziona Amazon

Questo argomento fornisce una panoramica del GameLift FlexMatch servizio Amazon, inclusi i componenti principali di un FlexMatch sistema e il modo in cui interagiscono.

Puoi utilizzarlo FlexMatch con giochi che utilizzano l'hosting GameLift gestito da Amazon o con giochi che utilizzano un'altra soluzione di hosting. I giochi ospitati sui GameLift server Amazon, inclusi i server in tempo reale, utilizzano il GameLift servizio Amazon integrato per individuare automaticamente i server di gioco disponibili e avviare le sessioni di gioco per le partite. I giochi che vengono utilizzati FlexMatch come servizio autonomo, incluso Amazon GameLift FleetIQ, devono 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 dei tuoi giochi, consulta [Nozioni di base su FlexMatch](#).

## Componenti del matchmaking

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

### GameLiftComponenti Amazon

Queste sono GameLift risorse Amazon che controllano il modo in cui il FlexMatch servizio esegue il matchmaking per il tuo gioco. Vengono creati e gestiti utilizzando GameLift gli strumenti di Amazon, tra cui la console e l'interfaccia a AWS riga di comando o, in alternativa, utilizzando l'AWSSDK per Amazon. GameLift

- FlexMatchconfigurazione del matchmaking (chiamato anche matchmaker) - Un matchmaker è un insieme di valori di configurazione che personalizza il processo di matchmaking per il tuo gioco. Un gioco può avere più matchmaker, ciascuno configurato per diverse modalità o esperienze di gioco, secondo necessità. Quando il tuo gioco invia una richiesta di matchmaking aFlexMatch, specifica quale matchmaker usare.
- FlexMatchset di regole di matchmaking: un set di regole contiene tutte le informazioni necessarie per valutare i giocatori per potenziali partite e approvarli o rifiutarli. 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, squadre o all'intera partita. Ad esempio, una regola potrebbe richiedere che tutti i giocatori della partita scelgano la stessa mappa di gioco, oppure che tutte le squadre abbiano una media di abilità dei giocatori simile.



- Coda delle sessioni di GameLift gioco di Amazon (solo per FlexMatch l'hosting GameLift gestito da Amazon): una coda di 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 cerca le risorse di hosting disponibili e come selezionare il miglior host disponibile per una partita.

## Componenti personalizzati

I seguenti componenti comprendono le funzionalità richieste per un FlexMatch sistema completo che devi implementare in base all'architettura del gioco.

- Interfaccia giocatore per il 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 il livello di abilità e i dati di latenza, se necessario per il processo di matchmaking.

### Note

Come buona prassi, la comunicazione con il FlexMatch servizio dovrebbe essere effettuata da un servizio di backend, non da un client di gioco.

- Servizio di matchmaking client: questo servizio invia le richieste di partecipazione del giocatore dall'interfaccia del giocatore, genera richieste di matchmaking e le invia al servizio. FlexMatch 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 tuo gioco, questo servizio può restituire ai giocatori le informazioni sulla connessione delle sessioni di gioco. Questo componente utilizza l'AWSSDK con l'GameLiftAPI Amazon per comunicare con il FlexMatch servizio.
- Servizio di collocamento delle partite (solo FlexMatch come servizio autonomo): questo componente funziona con il sistema di hosting di 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, inclusi ID dei giocatori, attributi e assegnazioni di squadra per tutti i giocatori in partita.

## FlexMatch processo di matchmaking

Questo argomento descrive uno scenario di matchmaking di base e le interazioni tra i vari componenti del gioco e il FlexMatch servizio.

## 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 tuo servizio di matchmaking client 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 sul giocatore richieste dal tuo matchmaker personalizzato, come il livello di abilità, le preferenze di gioco o i dati di latenza geografica. Puoi fare richieste di matchmaking per uno o più giocatori.

## Aggiungi richieste al pool di matchmaking

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

## 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 a partire 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 sono stati riempiti, 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 biglietto 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 equamente abbinate in base all'abilità del giocatore.

## Segnala i risultati del matchmaking

Quando viene trovata una corrispondenza accettabile, tutti i biglietti abbinati vengono aggiornati e viene generato un evento di matchmaking riuscito per ogni ticket abbinato.

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

### 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, inclusi gli ID dei giocatori e le assegnazioni delle squadre, quando impostano una sessione di gioco per la partita.

- FlexMatchcome servizio autonomo: il tuo servizio di posizionamento personalizzato ottiene i dati sui risultati delle partite da eventi di matchmaking di successo e si collega al tuo sistema di posizionamento delle sessioni di gioco esistente per individuare una risorsa di hosting disponibile per la partita. Una volta trovata una risorsa di hosting, il servizio di match placement si coordina con il sistema di hosting esistente per avviare una nuova sessione di gioco e acquisire informazioni sulla connessione.
- FlexMatchcon una soluzione di GameLift hosting Amazon: la coda delle sessioni 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 i dati sulla latenza del giocatore). Una volta completata con successo la sessione di gioco, il GameLift servizio Amazon richiede al server di gioco di avviare una nuova sessione di gioco, trasmettendo i risultati del matchmaking e altri dati di gioco opzionali.

### Connetti i giocatori alla partita

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

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

## FlexMatch supportato Regioni AWS

Se utilizzi una soluzione FlexMatch di GameLift hosting Amazon, puoi ospitare sessioni di gioco abbinate in qualsiasi luogo in cui offri giochi. Consulta l'[elenco completo Regioni AWS e le sedi per i GameLift hosting Amazon](#).

Per tutti FlexMatch gli utenti, puoi ospitare FlexMatch risorse, comprese le configurazioni di matchmaking e i set di regole, tra cui le seguenti sono supportate. Regioni AWS Per informazioni, consulta [Scegli una posizione per il matchmaker](#).

Nome Regione AWS	Codice regione
US East (N. Virginia)	us-east-1
US West (Oregon)	us-west-2
Asia Pacifico (Seul)	ap-northeast-2
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Cina (regioni di Pechino e Ningxia)	

# Configurazione di FlexMatch

Amazon GameLift FlexMatch è un AWS servizio e devi avere un AWS account per utilizzare questo servizio. Puoi creare un account AWS gratuitamente. Per ulteriori informazioni su cosa puoi fare con un AWS account, consulta [Guida introduttiva AWS](#).

Se utilizzi altre FlexMatch GameLift soluzioni Amazon, consulta i seguenti argomenti:

- [Configurazione dell'accesso per l'GameLifthosting Amazon e i server in tempo reale](#)
- [Configurazione dell'accesso per l'hosting su Amazon EC2 con Amazon GameLift FleetIQ](#)

Per configurare il tuo account per Amazon GameLift

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 del 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 alle tue FlexMatch risorse, utilizzando la GameLift console Amazon o chiamando le GameLift API di Amazon, devono avere accesso esplicito. Per istruzioni dettagliate sull'uso della console (o della AWS CLI o altri strumenti) per configurare gruppi di utenti, consulta [Creazione di utenti IAM](#).
3. Allega una politica sulle autorizzazioni al tuo utente o gruppo di utenti. L'accesso ai AWS servizi e alle risorse viene gestito allegando una [policy IAM](#) a un utente o a un gruppo di utenti. Le politiche sulle autorizzazioni specificano un insieme di AWS servizi e azioni a cui un utente deve avere accesso.

Per AmazonGameLift, devi 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 tutte le risorse e le azioni di AmazonGameLift. Puoi scegliere di limitare l'accesso specificando solo elementi FlexMatch -specifici.

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

```
"Statement":  
  {  
    "Effect": "Allow",  
    "Action": "gamelift:*",  
    "Resource": "*"  }  
}
```

# Nozioni di base su FlexMatch

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

## Argomenti

- [GameLiftFlexMatchIntegrazione con Amazon per il matchmaking autonomo](#)
- [FlexMatchintegrazione con Amazon GameLift hosting](#)

## GameLiftFlexMatchIntegrazione con Amazon per il matchmaking autonomo

Questo argomento delinea il processo di integrazione completo da implementare FlexMatch come servizio di matchmaking autonomo. Usa questa procedura se il tuo gioco multiplayer è ospitato utilizzando peer-to-peer hardware locale configurato su misura o altre primitive di cloud computing. Questo processo è utilizzabile anche con Amazon GameLift FleetIQ, una soluzione di ottimizzazione dell'hosting per giochi ospitati su Amazon EC2. Se stai ospitando il tuo gioco utilizzando l'hosting GameLift gestito da Amazon (inclusi i server in tempo reale), consulta [FlexMatchintegrazione con Amazon GameLift hosting](#).

Prima di iniziare l'integrazione, devi avere un AWS account e configurare le autorizzazioni di accesso per il GameLift servizio Amazon. Per informazioni dettagliate, consultare [Configurazione di FlexMatch](#). Tutte le attività essenziali relative alla creazione e alla gestione dei GameLift FlexMatch matchmaker e dei set di regole di Amazon possono essere eseguite utilizzando la GameLift console Amazon.

1. Crea un set di regole FlexMatch di matchmaking. Il tuo 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, che FlexMatch prevede l'inclusione o l'esclusione di giocatori da una partita. Questi requisiti potrebbero applicarsi ai singoli giocatori. Puoi anche personalizzare l'FlexMatch algoritmo nel set di regole, ad esempio per creare partite di grandi dimensioni con un massimo di 200 giocatori. Vedi questi argomenti:
  - [Crea un set di FlexMatch regole](#)
  - [FlexMatch esempi di set di regole](#)

2. Imposta le notifiche per gli eventi di matchmaking. Usa le notifiche per tenere traccia dell'attività di FlexMatch matchmaking, incluso lo stato delle richieste di partite 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'uso delle notifiche è l'opzione preferita a tal fine. Vedi questi argomenti:
  - [Configurare le notifiche FlexMatch degli eventi](#)
  - [FlexMatch eventi di matchmaking](#)
3. Configura una configurazione di FlexMatch matchmaking. Chiamato anche matchmaker, questo componente riceve richieste di matchmaking e le elabora. Puoi configurare un matchmaker specificando un set di regole, un obiettivo di notifica e il tempo massimo di attesa. Puoi anche abilitare le funzionalità opzionali. Vedi questi argomenti:
  - [Progetta un FlexMatch matchmaker](#)
  - [Crea una configurazione di matchmaking](#)
4. Crea un servizio di matchmaking per i clienti. Crea o espandi un servizio client di gioco con funzionalità a cui creare e inviare FlexMatch richieste di matchmaking. 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, le informazioni sulla latenza regionale. Deve inoltre disporre di un metodo per creare e assegnare ID ticket univoci per ogni richiesta. Puoi anche scegliere di creare un flusso di lavoro per l'accettazione dei giocatori che richieda ai giocatori di partecipare a una partita proposta. Questo servizio deve anche 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:
  - [Aggiungi FlexMatch a un client di gioco](#)
5. Crea un servizio di match placement. 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 avviare una nuova sessione di gioco per la partita. Potresti anche voler implementare un flusso di lavoro per effettuare richieste di backfill, che utilizza il matchmaking per riempire gli slot aperti nelle sessioni di gioco abbinata già in corso.




# FlexMatch integrazione con Amazon GameLift hosting

FlexMatch è disponibile con l'Amazon GameLift hosting gestito per server di gioco personalizzati e server in tempo reale. Per aggiungere l'abbinamento FlexMatch a un gioco, devi completare le attività riportate di seguito.

- 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:
    - [Crea un set di FlexMatch 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:
    - [Configurare le notifiche FlexMatch degli 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 matchmaker](#)
    - [Crea una configurazione di matchmaking](#)
- Integrare FlexMatch nel servizio del 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:
  - [Aggiungi FlexMatch a un client di gioco](#)
- Integrare FlexMatch nel 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:

- [Aggiungi FlexMatch a un server GameLift di gioco ospitato su Amazon](#)
- Configurare la funzionalità di backfill di FlexMatch (facoltativo). Richiedi ulteriori abbinamenti di giocatori per riempire gli slot giocatori aperti nei giochi esistenti. Puoi attivare il backfill automatico per fare in modo che Amazon GameLift gestisca 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

FlexMatchil backfill non è attualmente disponibile per i giochi che utilizzano server in tempo reale.

# Costruire un matchmaker Amazon GameLift FlexMatch

Un processo di abbinamento FlexMatch esegue la creazione di una partita per un gioco. Gestisce il pool di richieste di abbinamenti ricevute, crea le squadre per una partita, elabora e seleziona i giocatori per trovare i migliori gruppi di giocatori possibili e avvia il processo di collocazione e avvio di una sessione di gioco per la partita. Questo argomento descrive gli aspetti chiave di un matchmaker e come configurarne uno personalizzato per il proprio gioco.

Per una descrizione dettagliata sul modo in cui un matchmaker FlexMatch elabora le richieste di abbinamento ricevute, consulta la sezione [FlexMatch processo di matchmaking](#).

## Argomenti

- [Progetta un FlexMatch matchmaker](#)
- [Crea un set di FlexMatch regole](#)
- [Crea una configurazione di matchmaking](#)
- [Configurare le notifiche FlexMatch degli eventi](#)

## Progetta un FlexMatch matchmaker

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

## 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 [Crea un set di FlexMatch 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 [Configurare le notifiche FlexMatch degli 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 viene utilizzato FlexMatch con l'hosting GameLift gestito di Amazon, la coda della sessione 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 luoghi 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 gestisce un pool di ticket per le richieste di partite del tuo gioco, che vengono ordinate e valutate per individuare partite valide. Dopo aver creato una partita, Amazon GameLift 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 supportato Regioni AWS](#) i luoghi in cui puoi creare FlexMatch risorse.

Quando ne scegli uno 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 le richieste di FlexMatch 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.
- Se lo utilizzi FlexMatch con l'hosting GameLift gestito da Amazon, 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 utilizzi una soluzione FlexMatch di GameLift hosting Amazon, molte funzionalità sono

integrare. Se lo utilizzi 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 accettino 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

Usa il FlexMatch backfill per mantenere le sessioni di gioco piene di nuovi giocatori ben preparati per tutta la durata della sessione di gioco. Quando gestisci le richieste di backfill, FlexMatch utilizza lo stesso matchmaker utilizzato per abbinare i giocatori originali. Puoi personalizzare il modo in cui viene assegnata la 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 backfill manuale è disponibile indipendentemente dal fatto che il gioco venga utilizzato FlexMatch con l' GameLift hosting gestito di Amazon 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 backfill automatico è disponibile solo per i giochi che utilizzano l' GameLift hosting gestito di Amazon. Con questa funzionalità abilitata, se una sessione di gioco inizia con slot per giocatori aperti, Amazon GameLift inizia a generare automaticamente richieste di backfill relative. Questa funzione ti consente di configurare 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 i giochi che utilizzano FlexMatch l'hosting GameLift gestito da Amazon, puoi fornire informazioni aggiuntive da trasferire a un server di gioco ogni volta che viene richiesta una nuova sessione di gioco. Questo può essere un modo utile per eseguire le 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.

## Crea un set di FlexMatch regole

Ogni matchmaker FlexMatch deve disporre di 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 principianti 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 crei un set di regole, puoi utilizzare la GameLift console Amazon o l'interfaccia a AWS riga di comando.

### Argomenti

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

- [Crea set di regole di matchmaking](#)
- [FlexMatch esempi di set di regole](#)
- [FlexMatch linguaggio delle regole](#)

## Progetta un set di FlexMatch 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: definire la struttura e le dimensioni della squadra di una partita e dire 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, sono possibili le seguenti operazioni:

- Ottimizza l'algoritmo di matchmaking per il tuo gioco.
- Imposta requisiti minimi di latenza del giocatore per proteggere la qualità del gioco.
- Nel tempo, allenta gradualmente i requisiti della squadra e le regole delle partite 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.
- Elabora partite di grandi dimensioni con 40 o più giocatori. Per ulteriori informazioni sulla creazione di fiammiferi di grandi dimensioni, vedere [Progetta un FlexMatch set di regole ampio](#).

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

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

Puoi creare il tuo set di regole utilizzando la GameLift console Amazon 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 AmazonGameLift.
- **ruleLanguageVersion**— La versione del linguaggio delle espressioni delle proprietà utilizzata 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 partecipare i giocatori a partite accettabili con tempi di attesa minimi. Puoi personalizzare l'algoritmo e regolare il matchmaking per il tuo gioco.

Quello che segue è l'algoritmo di FlexMatch matchmaking predefinito:

1. FlexMatch inserisce tutti i ticket di matchmaking aperti e i ticket di riempimento in un pool di biglietti.
2. FlexMatch raggruppa casualmente i biglietti del pool in uno o più batch. Man mano che il pool di biglietti aumenta, FlexMatch forma lotti aggiuntivi per mantenere le dimensioni ottimali dei lotti.
3. FlexMatch ordina i biglietti per età, all'interno di ogni lotto.
4. FlexMatch crea una corrispondenza in base al ticket più vecchio di ogni batch.

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

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

- [Aggiungi l'ordinamento pre-batch](#)
- [Crea batch in base agli attributi BatchDistance](#)
- [Dai priorità ai ticket di riempimento](#)
- [Preferisci i biglietti più vecchi con espansioni](#)



## Aggiungi l'ordinamento pre-batch

Puoi ordinare il pool di ticket 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.

Definire i metodi di ordinamento pre-batch utilizzando la proprietà dell'algorithm.

`batchingPreference` L'impostazione predefinita è `random`.

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

- Ordina per attributi del giocatore. Fornisci un elenco degli attributi dei giocatori per preordinare il pool di ticket.

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

Nell'esempio seguente, FlexMatch ordina il pool di ticket in base alla mappa di gioco preferita dai giocatori e quindi in base all'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'algorithm `strategy subalanced`. La strategia bilanciata limita i tipi di dichiarazioni regolamentari disponibili. Per ulteriori informazioni, consulta [Progetta un FlexMatch set di regole ampio](#).

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

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

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

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

- La latenza accettabile corrisponde rapidamente. Il pool di ticket è preordinato in base alle località in cui i giocatori segnalano un valore di latenza accettabile. In questo modo si ottengono meno lotti contenenti più ticket. 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 backfill automatico o il backfill manuale, puoi personalizzare il modo in cui FlexMatch elabora i ticket di matchmaking in base al tipo di richiesta. Il tipo di richiesta può essere una nuova richiesta di corrispondenza o di riempimento. Per impostazione predefinita, FlexMatch tratta entrambi i tipi di richieste allo stesso modo.

La prioritizzazione del backfill influisce sulla gestione dei ticket dopo averli FlexMatch raggruppati in batch. La prioritizzazione del backfill richiede set di regole per utilizzare una strategia di ricerca esaustiva.

FlexMatch non corrisponde a più ticket di riempimento contemporaneamente.

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

- Abbina prima i biglietti di riempimento. Questa opzione cerca di abbinare i ticket di riempimento 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 backfill automatico viene spesso utilizzato nei giochi con sessioni di gioco brevi e un elevato turnaround dei giocatori. Il backfill automatico aiuta questi giochi a formare un numero minimo di partite possibili e ad avviarle mentre FlexMatch cercano più giocatori per riempire gli slot liberi.

Imposta `backfillPriority` su `high`.

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

- Abbina i biglietti di riserva per ultimi. Questa opzione ignora i ticket di riempimento finché non valuta tutti gli altri ticket. Ciò significa che FlexMatch inserisce i giocatori in entrata nei giochi esistenti quando non riesce ad abbinarli a nuovi giochi.

Questa opzione è utile quando vuoi usare 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 espansioni

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

Per modificare quando vengono FlexMatch applicate le regole di espansione, impostate 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 che FlexMatch corrisponde un nuovo biglietto, l'ora viene azzerata. Con questa opzione, le partite risultanti tendono ad essere di qualità superiore ma richiedono più tempo per essere abbinati; le richieste di corrispondenza potrebbero scadere prima del completamento se impiegano troppo tempo a corrispondere. `expansionAgeSelection` impostato su `newest`. `newest` è l'impostazione predefinita.
- Espandi in base ai biglietti più vecchi. Questa opzione applica le regole di espansione in base al 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. Ci sono 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 di partita. Ad esempio, potresti voler passare le scelte dei personaggi 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` o `mappa delle stringhe`.

- **default** (opzionale): inserisci 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, non è FlexMatch possibile soddisfare la richiesta.

## Definisci le squadre di gara

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 abbinamento come abbinamento piccolo o grande, in base a come il set di regole definisce le dimensioni del team. Le partite potenziali con un massimo di 40 giocatori sono partite piccole, le partite con più di 40 giocatori sono partite grandi. 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 team un nome univoco. Usi questo nome nelle regole e nelle espansioni e nei 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 creare con questa definizione. Quando FlexMatch crea una partita, assegna a queste squadre il nome fornito con un numero aggiunto. Ad esempio `Red-Team1,Red-Team2, eRed-Team3`.

FlexMatch tenta di riempire le squadre fino alla dimensione massima dei 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 abbinare i 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 elabora una richiesta di partita, parte dal giocatore più anziano nel pool di giocatori disponibili e costruisce una partita attorno a quel giocatore. Per una guida dettagliata sulla creazione di FlexMatch regole, consulta [FlexMatch tipi di regole](#).

- **nome (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):** utilizza 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 [FlexMatchlinguaggio 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. Ulteriori informazioni sulle proprietà e su come utilizzare il linguaggio di espressione delle proprietà FlexMatch sono disponibili in [FlexMatchlinguaggio delle regole](#).

## Consenti ai requisiti di attenuarsi nel tempo

Le espansioni ti consentono di allentare i criteri delle regole nel tempo quando non FlexMatch riesci a trovare una corrispondenza. Questa funzione FlexMatch garantisce che il meglio sia disponibile quando non è possibile creare un abbinamento perfetto. Allentando le regole con un'espansione, allargherai gradualmente il numero di giocatori che possono contare su una partita accettabile.

Le espansioni iniziano quando l'età del biglietto più recente della partita incompleta corrisponde al tempo di attesa dell'espansione. Quando FlexMatch aggiungi un nuovo biglietto alla partita, il tempo di attesa dell'espansione potrebbe essere resettato. Puoi personalizzare la modalità di avvio 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 dichiarazione sulla distanza, chiamata `SkillDelta` per richiedere che tutti i giocatori in una partita si trovino a meno di 5 livelli di abilità l'uno dall'altro. Se non vengono effettuate nuove partite per quindici secondi, questa espansione cerca una differenza di livello di abilità di 10, quindi dieci secondi dopo cerca una differenza di 20.

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

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

Con i matchmaker che hanno il riempimento automatico abilitato, non abbassare i requisiti di conteggio dei giocatori troppo velocemente. 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. 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 ampio

Se il tuo set di regole crea partite che consentono da 41 a 200 giocatori, devi apportare alcune modifiche alla configurazione del set di regole. Queste regolazioni 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 di abbinamento 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 set di regole di partita ampio.

Per ottimizzare il set di regole per partite di grandi dimensioni, apporta le modifiche descritte di seguito. Consulta lo schema per un set di regole di corrispondenza di grandi dimensioni [Schema del set di regole per partite di grandi dimensioni](#) ed esempi di set di regole in [Esempio 7: crea una partita grande](#).

Personalizza l'algoritmo di abbinamento per partite di grandi dimensioni

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

- `strategy`(obbligatorio) — Imposta la `strategy` proprietà su «bilanciata». Questa impostazione attiva l'esecuzione FlexMatch di ulteriori controlli post-partita per trovare il bilanciamento ottimale della squadra in base a uno specifico attributo del giocatore, definito nella proprietà `balancedAttribute`. 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 le abilità dei giocatori, 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 formazione delle 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 posizione 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 giochi 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».
  - La posizione più veloce. FlexMatch preordina tutti i ticket nel pool in base a dove riportano i valori di latenza più bassi. Di conseguenza, le partite tendono a essere formate con giocatori che segnalano una bassa latenza nelle stesse località. Allo stesso tempo, il potenziale parco 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 notevolmente per quanto riguarda l'attributo di bilanciamento.

L'esempio seguente configura l'algoritmo delle partite in modo che si comporti come segue: (1) Preordina il pool di ticket per raggruppare i ticket per località in cui hanno valori di latenza accettabili; (2) Crea batch di biglietti ordinati per gli abbinamenti; (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 del giocatore 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 giocatore, ma il bilanciamento degli attributi funziona meglio quando vengono forniti valori specifici per il giocatore.

## Definisci i team

Il processo di definizione delle dimensioni e della struttura del team è lo stesso degli abbinamenti piccoli, ma il modo in cui FlexMatch completa i team è diverso. Ciò influisce sull'aspetto delle partite se riempite solo parzialmente. In risposta, potresti voler modificare le dimensioni minime del team.

FlexMatch utilizza le seguenti regole quando si assegna un giocatore a un team. 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 quasi uguale di giocatori, 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 aperti. 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 hanno ora raggiunto `minPlayers`.
2. La squadra verde non è ancora arrivata a `minPlayers`. I successivi 2 giocatori vengono assegnati al team verde. La squadra verde ha ora raggiunto `minPlayers`.
3. Con tutte le squadre presentando `minPlayers`, ora vengono assegnati giocatori aggiuntivi in base al numero di slot aperti. Le squadre rosse e blu hanno ciascuna 5 posti liberi, mentre la squadra verde ne ha 1. I successivi 8 giocatori vengono assegnati (4 ciascuno) alle squadre Rossa e Blu. Tutte le squadre ora hanno 1 posto libero.
4. Gli slot rimanenti per 3 giocatori vengono assegnati (1 ciascuno) alle squadre senza un ordine particolare.

## Stabilisci 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 un attributo del giocatore specificato. Questo è diverso dalla definizione di un attributo di bilanciamento come parte dell'algoritmo delle partite larghe, che si concentra sulla creazione di squadre equamente abbinata. Questa regola raggruppa i ticket di matchmaking in base alla somiglianza nei valori degli attributi specificati, come l'abilità principiante o esperto, il che tende a far sì che i giocatori 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 [Regola della distanza tra batch](#) riferimento. Ecco un esempio che richiede che i giocatori di una partita si trovino a un livello di abilità l'uno dall'altro:

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

## Rilassati sui requisiti delle partite di grandi

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. Con partite di grandi dimensioni, hai la possibilità di allentare le regole di latenza o il conteggio dei giocatori di squadra.

Se utilizzi il riempimento automatico delle partite per partite di grandi dimensioni, evita di abbassare il conteggio dei giocatori della tua squadra troppo velocemente. FlexMatch inizia a generare richieste di backfill solo dopo l'inizio di una sessione di gioco, cosa che potrebbe non avvenire per diversi secondi dopo la creazione di una partita. Durante tale periodo, FlexMatch può creare più nuove sessioni di gioco parzialmente completate, specialmente quando le regole del numero 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
  }]
}]
```

## Crea set di regole di matchmaking

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

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.

### Argomenti correlati

- [Progetta un set di FlexMatch regole](#)
- [FlexMatch esempi di set di regole](#)
- [FlexMatch linguaggio delle regole](#)

## Console

### Crea un set di regole

1. Apri la GameLift console Amazon all'[indirizzo https://console.aws.amazon.com/gamelift/](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 set di FlexMatch 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 monitorare AWS le tue risorse.
6. Scegli 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 restituisce un [MatchmakingRuleSet](#) oggetto che include le impostazioni che hai specificato. Un matchmaker può ora utilizzare il nuovo set di regole.

## Console

### Eliminazione di un set di regole

1. Apri la GameLift console Amazon all'[indirizzo https://console.aws.amazon.com/gamelift/](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 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 restituisce un messaggio di errore. In tal caso, 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-configuration](#) e 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 alla struttura di FlexMatch configurazione e al linguaggio delle espressioni delle proprietà. Copiare questi set di regole interamente o scegliere i componenti in base alle esigenze.

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

- [Crea un set di FlexMatch regole](#)
- [Progetta un set di FlexMatch regole](#)
- [FlexMatchschema del set di regole](#)
- [FlexMatchlinguaggio delle regole](#)

### Note

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

## Esempio 1: 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 2: 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 3: 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 abbinare in modo uniforme in base alle competenze dei giocatori. Per valutare la `FairTeamSkill` regola, aggiunge FlexMatch 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 4: utilizza l'ordinamento esplicito per trovare gli abbinamenti migliori

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 5: trovare le intersezioni degli 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 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 6: 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 7: crea una partita grande

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 ampio](#).

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 [Rilassati sui requisiti delle partite di grandi](#).

```
{
  "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 8: 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 le partite secondo questo set di regole prima che qualsiasi espansione abbia 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 [Rilassati sui requisiti delle partite di grandi](#).

```
{
  "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 9: Crea una partita di grandi dimensioni con giocatori con attributi 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 10:** 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 utilizzando 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 11: crea una regola che utilizzi la lista di blocco 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 ID dei giocatori (fino a 100).
- Confronta tutti i giocatori con la lista di blocco di ogni giocatore e rifiuta una partita proposta se vengono trovati ID giocatore bloccati.

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 GameLift FlexMatch matchmaker Amazon per elaborare le richieste di matchmaking, crea una configurazione di matchmaking. Usa la GameLift console Amazon o il AWS Command Line Interface (AWS CLI). Per ulteriori informazioni sulla creazione di un matchmaker, vedere [Progetta un FlexMatch matchmaker](#).

## Crea un matchmaker per l'hosting Amazon GameLift

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

### Console

1. Nella [GameLiftconsole Amazon](#), 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 di matchmaking.
4. Nella pagina Definisci i dettagli di configurazione, in Dettagli di configurazione del matchmaking, procedi come segue:
  - a. Per Nome, inserisci un nome del matchmaker che possa aiutarti a identificarlo in un elenco e nelle metriche. Il nome del matchmaker deve essere univoco all'interno della

- Regione. Le richieste di matchmaking identificano quale matchmaker utilizzare in base al nome e alla regione.
- b. (Facoltativo) Per Descrizione, aggiungi una descrizione per aiutare a identificare il matchmaker.
  - c. Per Set di regole, scegli un set di regole dall'elenco da utilizzare con il matchmaker. L'elenco contiene tutti i set di regole che hai creato nella regione corrente.
  - d. Per la FlexMatchmodalità, scegli Managed for Amazon GameLift managed hosting. Questa modalità richiede di FlexMatch passare le partite riuscite alla coda della sessione di gioco specificata.
  - e. Per AWSRegione, scegli la regione in cui hai configurato la coda delle sessioni di gioco che desideri utilizzare con il matchmaker.
  - f. In Coda, scegli la coda della sessione di gioco che desideri utilizzare con il matchmaker.
5. Seleziona Successivo.
6. Nella pagina Configura impostazioni, in Impostazioni matchmaking, procedi come segue:
- a. Per il timeout della richiesta, imposta il tempo massimo, in secondi, in cui il matchmaker deve completare una partita per ogni richiesta. FlexMatchannulla le richieste di matchmaking che superano questo tempo.
  - b. Per la modalità Backfill, scegli una modalità per gestire i backfill delle corrispondenze.
    - 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 aggiuntivi, imposta il numero di slot da tenere aperti in una partita. FlexMatchpuò riempire questi slot con giocatori in futuro.
  - d. (Facoltativo) In Opzioni di accettazione della partita, in 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, per il timeout di accettazione, imposta per quanto tempo, in secondi, vuoi che il matchmaker attenda l'accettazione 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 di Amazon Simple Notification Service (Amazon SNS) per ricevere le notifiche degli eventi di matchmaking. Se non hai ancora impostato un argomento SNS, puoi sceglierlo in seguito modificando la



- configurazione del matchmaking. Per ulteriori informazioni, consulta [Configurare le notifiche FlexMatch degli eventi](#).
- b. (Facoltativo) Per i dati personalizzati degli eventi, inserisci 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 delle sessioni di gioco, inserisci eventuali informazioni aggiuntive relative al gioco che desideri fornire FlexMatch alle nuove sessioni di gioco iniziate con le partite realizzate 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 AWS delle tue risorse.
  10. Seleziona Successivo.
  11. Nella pagina Rivedi e crea, rivedi le tue scelte, quindi scegli Crea. In caso di creazione riuscita, il matchmaker è pronto ad accettare le richieste di matchmaking.

## AWS CLI

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

Questo comando di esempio crea una nuova configurazione di matchmaking che richiede l'accettazione del giocatore e abilita il backfill automatico. Inoltre, riserva slot per due giocatori per FlexMatch aggiungere giocatori in un secondo momento e fornisce alcuni dati sulle sessioni 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 restituisce un [MatchmakingConfiguration](#) oggetto con le impostazioni richieste per il matchmaker. Il nuovo matchmaker è pronto ad accettare le richieste di matchmaking.

## 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 la GameLift console Amazon all'[indirizzo https://console.aws.amazon.com/gamelift/home](https://console.aws.amazon.com/gamelift/home).
2. Passa alla AWS regione in cui desideri creare il tuo matchmaker. Per un elenco di regioni che supportano le configurazioni di FlexMatch matchmaking, vedere. [Scegli una posizione per il matchmaker](#)
3. Nel riquadro di navigazione, scegli FlexMatch, Configurazioni di matchmaking.
4. Nella pagina delle configurazioni del matchmaking, scegli Crea configurazione di matchmaking.
5. Nella pagina Definisci i dettagli di configurazione, in Dettagli di configurazione del matchmaking, procedi come segue:
  - a. Per Nome, inserisci un nome del matchmaker che possa aiutarti a identificarlo in un elenco e nelle metriche. Il nome del matchmaker deve essere univoco all'interno della Regione. Le richieste di matchmaking identificano quale matchmaker utilizzare in base al nome e alla regione.
  - b. (Facoltativo) Per Descrizione, aggiungi una descrizione per aiutare a identificare il matchmaker.
  - c. Per Set di regole, scegli un set di regole dall'elenco da utilizzare con il matchmaker. L'elenco contiene tutti i set di regole che hai creato nella regione corrente.

- d. Per la FlexMatchmodalità, scegli Standalone. Ciò indica che disponi di un meccanismo personalizzato per avviare nuove sessioni di gioco su una soluzione di hosting esterna ad AmazonGameLift.
6. Seleziona Successivo.
7. Nella pagina Configura impostazioni, in Impostazioni matchmaking, procedi come segue:
  - a. Per il timeout della richiesta, imposta il tempo massimo, in secondi, in cui il matchmaker deve completare una partita per ogni richiesta. Le richieste di matchmaking che superano questo tempo vengono rifiutate.
  - b. (Facoltativo) In Opzioni di accettazione della partita, in 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, per il timeout di accettazione, imposta per quanto tempo, in secondi, vuoi che il matchmaker attenda l'accettazione 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 di eventi di matchmaking. Se non hai ancora impostato un argomento SNS, puoi sceglierlo in seguito modificando la configurazione del matchmaking. Per ulteriori informazioni, consulta [Configurare le notifiche FlexMatch degli eventi](#).
  - b. (Facoltativo) Per i dati personalizzati degli eventi, inserisci i dati personalizzati che desideri associare a questo matchmaker nella messaggistica degli eventi. FlexMatchinclude questi dati in ogni evento associato al matchmaker.
9. (Facoltativo) In Tag, aggiungi tag per aiutarti a gestire e tenere traccia AWS delle tue risorse.
10. Seleziona Successivo.
11. Nella pagina Rivedi e crea, rivedi le tue scelte, quindi scegli Crea. In caso di creazione riuscita, il matchmaker è pronto ad accettare le richieste di matchmaking.

## AWS CLI

Per creare una configurazione di abbinamento con AWS CLI, apri una finestra a riga di comando e utilizza il comando [create-matchmaking-configuration](#) 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 "SampleMatchmaker123" \  
  --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 restituisce un [MatchmakingConfiguration](#) oggetto con le impostazioni richieste per il matchmaker. Il nuovo matchmaker è pronto ad accettare le richieste di matchmaking.

## Modifica una configurazione di matchmaking

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

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

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

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

## Configurare le notifiche FlexMatch degli eventi

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

Sono disponibili due opzioni per impostare le notifiche di eventi.

- Chiedi al tuo matchmaker di pubblicare notifiche di 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 degli FlexMatch eventi GameLift emessi da Amazon, consulta [FlexMatch eventi di matchmaking](#).

## Configurare EventBridge eventi

Amazon pubblica GameLift automaticamente tutti gli eventi di matchmaking su AmazonEventBridge. ConEventBridge, 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 [Cos'è AmazonEventBridge?](#)

### Note

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

## Configura un argomento Amazon SNS

Puoi fare in modo che Amazon GameLift pubblichi tutti gli eventi generati da un FlexMatch matchmaker su un argomento Amazon SNS.

Per creare un argomento SNS per le notifiche GameLift degli eventi Amazon

1. Aprire 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 [Per creare un argomento utilizzando la Guida per AWS Management Console](#) gli sviluppatori di Amazon Simple Notification Service.
5. Nella pagina Dettagli relativa all'argomento, scegli Modifica.

6. (Facoltativo) Nella pagina Modifica relativa all'argomento, espandi Access policy, quindi aggiungi la sintassi in grassetto della seguente dichiarazione politica AWS Identity and Access Management (IAM) alla fine della policy 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 la configurazione di Amazon GameLift 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. Scegliere Save changes (Salva modifiche).

## Configura 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 Amazon SNS utilizzando chiavi gestite in AWS Key Management Service (AWS KMS). Per ulteriori informazioni sulla crittografia lato server con Amazon SNS, consulta Encryption [at rest](#) nella Amazon Simple Notification Service Developer Guide.

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

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

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

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "gamelift.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey"
  ],
  "Resource": "*",
  "Condition": {
    "ArnLike": {
```

```
    "aws:SourceArn":
      "arn:aws:gamelift:your_region:your_account:matchmakingconfiguration/your_configuration_name"
    },
    "StringEquals": {
      "kms:EncryptionContext:aws:sns:topicArn":
        "arn:aws:sns:your_region:your_account:your_sns_topic_name"
    }
  }
}
```

## Configurare una sottoscrizione agli argomenti per richiamare una funzione Lambda

Puoi richiamare una funzione Lambda utilizzando le notifiche degli eventi pubblicate nel 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 `MyFlexMatchEventTopic` per richiamare una funzione Lambda denominata `FlexMatchEventHandlerLambdaFunction`. Il modello crea una politica di autorizzazioni IAM che consente GameLift ad Amazon di 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
```

# Preparazione di giochi per FlexMatch

Usa Amazon GameLift FlexMatch per aggiungere la funzionalità di matchmaking dei giocatori ai tuoi giochi. FlexMatch è disponibile con le GameLift soluzioni Amazon gestite per server di gioco personalizzati e server in tempo reale.

FlexMatch associa il servizio di abbinamento a un motore di regole personalizzabili. Ciò ti consente di progettare le modalità di abbinamento dei giocatori in base ai relativi attributi e alle modalità più adatte per il tuo gioco e di fare affidamento su FlexMatch per la gestione degli aspetti tecnici della formazione di gruppi di giocatori e del loro posizionamento nei giochi. Vedi maggiori dettagli sul matchmaking personalizzato in [FlexMatch esempi di set di regole](#).

FlexMatch crea la caratteristica Code. Una volta formato un abbinamento, FlexMatch invia i relativi dettagli a una coda di tua scelta. La coda cerca le risorse di hosting disponibili sulle tue GameLift flotte Amazon e avvia una nuova sessione di gioco per la partita.

Gli argomenti di questa sezione descrivono come aggiungere il supporto dell'abbinamento ai server e ai client di gioco. Per creare un matchmaker per il tuo gioco, consulta [Costruire un matchmaker Amazon GameLift FlexMatch](#). Per ulteriori informazioni sul funzionamento di FlexMatch, consulta [Come GameLift FlexMatch funziona Amazon](#).

## Aggiungi FlexMatch a un client di gioco

Questo argomento descrive come aggiungere il supporto per il FlexMatch matchmaking ai tuoi servizi di gioco lato client. Il processo è essenzialmente lo stesso indipendentemente dal fatto che tu lo utilizzi FlexMatch con l'hosting GameLift gestito di Amazon o con un'altra soluzione di hosting. Per ulteriori informazioni su FlexMatch e su come impostare un matchmaker personalizzato per i tuoi giochi, consulta i seguenti argomenti:

- [FlexMatch integrazione con Amazon GameLift hosting](#)
- [Come GameLift FlexMatch funziona Amazon](#)
- [Costruire un matchmaker Amazon GameLift FlexMatch](#)
- [FlexMatch esempi di set di regole](#)

Per abilitare il FlexMatch matchmaking nel tuo gioco, aggiungi le seguenti funzionalità:

- Preparati a richiedere il matchmaking per uno o più giocatori (richiesto).
- Tieni traccia dello stato delle richieste di matchmaking (richiesto).
- Richiedi l'accettazione del giocatore per una partita proposta (opzionale).
- Dopo aver creato una sessione di gioco per la nuova partita, ottieni le informazioni sulla connessione del giocatore e partecipa al gioco.

## Preparati a richiedere il matchmaking per i giocatori

Consigliamo vivamente al tuo client di gioco di effettuare richieste di matchmaking tramite un servizio di gioco lato client. Utilizzando una fonte attendibile, puoi proteggerti più facilmente dai tentativi di hacking e dai dati di giocatori falsi. Se il gioco ha un servizio di directory di sessione, questa è una buona opzione per gestire le richieste di abbinamento.

Per preparare il servizio client, esegui le attività seguenti:

- Aggiungi l'GameLiftAPI Amazon. Il tuo servizio clienti utilizza le funzionalità dell'GameLiftAPI Amazon, che fa parte dell'AWSSDK. Consulta [GameLift gli SDK Amazon per i servizi clienti per saperne di più sull'AWSSDK](#) e scaricare la versione più recente. Aggiungi questo SDK al progetto del servizio client di gioco.
- Configura un sistema di ticket di matchmaking. A tutte le richieste di abbinamento deve essere assegnato un ID ticket univoco. È necessario un meccanismo per generare gli ID univoci e assegnarli alle nuove richieste di abbinamento. Un ID ticket può utilizzare qualsiasi formato di stringa, fino a un massimo di 128 caratteri.
- Ottieni le informazioni del matchmaker. Ottieni il nome della configurazione di abbinamento che pensi di utilizzare. È inoltre necessario l'elenco del matchmaker di attributi giocatore obbligatori, definiti nel set di regole del matchmaker.
- Ottenere i dati dei giocatori. Configura un modo per ottenere i dati rilevanti per ogni giocatore. Questo include l'ID giocatore, i valori degli attributi dei giocatori e i dati di latenza aggiornati per ogni regione in cui il giocatore è probabilmente inserito in un gioco.
- (facoltativo) Abilita il backfill degli abbinamenti. Decidi come eseguire il backfill dei tuoi giochi abbinati esistenti. Se i matchmaker hanno la modalità backfill impostata su "manuale", si potrebbe voler aggiungere il supporto backfill al gioco. Se la modalità backfill è impostata su "automatico", potrebbe essere necessario un modo per disattivarlo per le singole sessioni di gioco. Ulteriori informazioni sulla gestione del backfill degli abbinamenti sono disponibili in [Riempi i giochi esistenti con FlexMatch](#).

## Richiedi il matchmaking per i giocatori

Aggiungi il codice al servizio client per creare e gestire le richieste di abbinamento a un matchmaker FlexMatch. Il processo di richiesta del FlexMatch matchmaking è identico per i giochi che utilizzano FlexMatch l'GameLiftmanagedhosting Amazon e per i giochi che utilizzano FlexMatch come soluzione autonoma.

Crea una richiesta di abbinamento:

- Chiama l'GameLiftAPI Amazon [StartMatchmaking](#). Ogni richiesta deve contenere le seguenti informazioni.

### Matchmaker

Il nome della configurazione del matchmaking da utilizzare per la richiesta.

FlexMatch inserisce ogni richiesta nel pool per il matchmaker specificato e la richiesta viene elaborata in base alla configurazione del 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 matchmaker](#).

### 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 tenta di creare una singola corrispondenza 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, che viene generato. Vedi [Generare gli ID dei giocatori](#).
- 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 [Crea un set di FlexMatch 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.

Recupera i dettagli della richiesta di abbinamento:

- Una volta inviata una richiesta di partita, 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.

Annulla 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

Configura le notifiche per tenere traccia degli eventi che Amazon GameLift emette per i processi di matchmaking. Puoi configurare le notifiche direttamente, creando un argomento SNS o utilizzando AmazonEventBridge. Per ulteriori informazioni su come impostare le notifiche, consulta [Configurare le notifiche FlexMatch degli 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. La procedura di gestione delle accettazioni da parte dei giocatori è identica per i giochi che utilizzano FlexMatch l'hosting GameLift gestito da Amazon 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 l'`FlexMatcheventoMatchmakingRequiresAcceptance`.
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. Riportare 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 dei giocatori che hanno accettato la partita vengono automaticamente restituiti alla biglietteria. I biglietti per i giocatori che non hanno accettato la partita passano allo stato `FALLIMENTO` e non vengono più processati. Per i biglietti con più giocatori, se uno dei giocatori presenti nel ticket non ha accettato la partita, l'intero ticket viene considerato invalido.

## Connettiti a una partita

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

Per i giochi che utilizzano l'hosting GameLift gestito da Amazon, quando una richiesta di matchmaking viene soddisfatta con successo, le informazioni di connessione della 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 della partita della sessione di gioco, che includono le assegnazioni della squadra (vedi [GameSession](#)).

Per i giochi che utilizzano altre soluzioni di hosting, incluso Amazon GameLift FleetIQ, devi integrare un meccanismo che consenta ai giocatori in partita di connettersi alla sessione di gioco appropriata.

## Esempi di richieste di matchmaking

I seguenti frammenti di codice generano 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)
```

## Aggiungi FlexMatch a un server GameLift di gioco ospitato su Amazon

Questo argomento descrive come aggiungere il supporto per il FlexMatch matchmaking ai server di gioco personalizzati che utilizzano l'hosting GameLift gestito da Amazon. Per ulteriori informazioni sull'aggiunta di FlexMatch ai tuoi giochi, consulta i seguenti argomenti:

- [Come GameLift FlexMatch funziona Amazon](#)
- [FlexMatch integrazione con Amazon GameLift hosting](#)

Le informazioni in questo argomento presuppongono che tu abbia integrato correttamente Amazon GameLift Server SDK nel tuo progetto di server di gioco, come descritto in [Aggiungi Amazon GameLift al tuo server di gioco](#). Al termine di questa procedura, disporrai della maggior parte dei

meccanismi necessari. Nelle sezioni di questo argomento vengono descritte le procedure rimanenti per gestire i giochi configurati con FlexMatch.

## Configura il tuo server di gioco per il matchmaking

Per configurare il server di gioco sulla gestione dei giochi abbinati, completa le attività seguenti.

1. Avviare le sessioni di gioco create tramite il processo di abbinamento. Per richiedere una nuova sessione di gioco, Amazon GameLift invia una `onStartGameSession()` richiesta al tuo server di gioco con un oggetto della sessione di gioco (vedi [GameSession](#)). Il server di gioco utilizza le informazioni sulla sessione di gioco, inclusi i dati personalizzati sul gioco, per avviare la sessione di gioco richiesta. Per maggiori dettagli, consulta [Avvio di una sessione di gioco](#).

Per i giochi abbinati, l'oggetto della sessione di gioco contiene anche un set di dati del matchmaker. I dati del matchmaker includono le informazioni che il server di gioco richiede per avviare una nuova sessione di gioco per l'abbinamento. Sono inclusi la struttura del team di abbinamento, le assegnazioni al team e determinati attributi di giocatore che potrebbero essere importanti per il gioco. Ad esempio, il gioco potrebbe sbloccare determinate funzionalità o livelli in base al livello medio di abilità del giocatore o scegliere una mappa in base alle preferenze dei giocatori. Ulteriori informazioni in [Lavora con i dati del matchmaker](#).

2. Gestire le connessioni dei giocatori. Quando ti connetti a una partita abbinata, un client di gioco fa riferimento a un ID giocatore e a un ID di sessione del giocatore (vedi [Convalidare un nuovo giocatore](#)). Il tuo server di gioco utilizza l'ID giocatore per associare un giocatore in arrivo con le informazioni del giocatore nei dati del matchmaker. I dati del matchmaker identificano l'assegnazione al team di un giocatore e possono fornire altre informazioni per rappresentare in modo corretto il giocatore nel gioco.
3. Segnalare quando i giocatori escono dal gioco. Assicurati che il tuo server di gioco stia chiamando l'API del server `RemovePlayerSession()` per segnalare i giocatori abbandonati (vedi [Segnalare la fine della sessione di un giocatore](#)). Questa fase è importante se si utilizza il backfill di FlexMatch per riempire gli slot vuoti in giochi esistenti. È fondamentale se il gioco avvia richieste di backfill tramite un servizio di gioco sul lato client. Per ulteriori informazioni sull'implementazione del backfill di FlexMatch, consultare la sezione [Riempi i giochi esistenti con FlexMatch](#).
4. Richiedere nuovi giocatori per le sessioni di gioco abbinata esistenti (facoltativo). Decidi come eseguire il backfill dei tuoi giochi abbinati esistenti. Se i matchmaker hanno la modalità backfill impostata su "manuale", si potrebbe voler aggiungere il supporto backfill al gioco. Se la modalità backfill è impostata su "automatico", potrebbe essere necessario un modo per disattivarlo per

le singole sessioni di gioco. Ad esempio, è possibile arrestare il backfill di una sessione di gioco una volta raggiunto un determinato punto del gioco. Ulteriori informazioni sulla gestione del backfill degli abbinamenti sono disponibili in [Riempi i giochi esistenti con FlexMatch](#).

## Lavora con i dati del matchmaker

Il tuo server di gioco deve essere in grado di riconoscere e utilizzare le informazioni di gioco in un [GameSession](#) oggetto. Il GameLift servizio Amazon trasmette questi oggetti al tuo server di gioco ogni volta che viene avviata o aggiornata una sessione di gioco. Le informazioni principali della sessione di gioco includono l'ID e il nome della sessione di gioco, il numero massimo di giocatori, le informazioni di connessione e i dati di gioco personalizzati (se specificati).

Per le sessioni di gioco create tramite FlexMatch, l'oggetto `GameSession` contiene anche un set di dati del matchmaker. Oltre a un ID di abbinamento univoco, identifica il matchmaker che ha creato l'abbinamento e descrive i team, le assegnazioni al team e i giocatori. Include gli attributi giocatore della richiesta di abbinamento originale (vedi l'oggetto [Player](#)). Non include la latenza del giocatore; se sono necessari i dati sulla latenza dei giocatori correnti, ad esempio per il backfill degli abbinamenti, ti consigliamo di procurarti dei dati aggiornati.

### Note

I dati di Matchmaker specificano la configurazione completa del matchmaking ARN, che identifica il nome della configurazione, AWS l'account e la regione. Quando si richiede il backfill degli abbinamenti da un client o un servizio di gioco, è necessario solo il nome della configurazione. Puoi estrarre il nome della configurazione analizzando la stringa che segue `":matchmakingconfiguration/"`. Nell'esempio mostrato, il nome della configurazione del matchmaking è `"MyMatchmakerConfig"`.

Il codice JSON seguente mostra un insieme di dati del matchmaker tipico. Questo esempio illustra un gioco con due giocatori, con giocatori abbinati in base alle valutazioni della competenza e al livello più alto raggiunto. Il matchmaker effettua gli abbinamenti anche in base ai personaggi e verifica che i giocatori abbinati dispongano di almeno una preferenza di mappatura in comune. In questo scenario, il server di gioco deve essere in grado di determinare quale mappa è la preferita e utilizzarla nella sessione di gioco.

```
{
```

```
"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
            }
          }
        }
      }
    ]
  }
]
```

## Riempi i giochi esistenti con FlexMatch

Il backfill degli abbinamenti utilizza il meccanismo FlexMatch per trovare nuovi giocatori per le sessioni di gioco abbinata esistenti. Anche se puoi sempre aggiungere giocatori a qualsiasi gioco (vedi [Partecipare a una sessione di gioco](#)), il riempimento delle partite garantisce che i nuovi giocatori soddisfino gli stessi criteri di abbinamento 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 server in tempo reale.

Esistono due tipi di meccanismi di backfill:

- Per riempire le sessioni di gioco che iniziano con un numero di giocatori inferiore al numero massimo consentito, abilita il backfill automatico.
- Per sostituire i giocatori che abbandonano una sessione di gioco in corso, aggiungi funzionalità al server di gioco per inviare richieste di backfill.

## Attiva il riempimento automatico

Con il backfill automatico delle partite, Amazon attiva GameLift automaticamente una richiesta di riempimento ogni volta che una sessione di gioco inizia con uno o più slot per giocatori vuoti. 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 avvia 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 inizia a generare richieste di backfill 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 alla compilazione delle richieste di riempimento prima di creare nuove partite. Nel tuo set di regole di matchmaking, aggiungi un componente dell'algoritmo e imposta la priorità di backfill su «alta». Per ulteriori dettagli, consulta [Personalizza l'algoritmo di abbinamento](#).
3. Aggiorna la sessione di gioco con nuovi dati del matchmaker. Amazon GameLift aggiorna il tuo server di gioco con le informazioni sulle partite utilizzando la funzione di callback del Server SDK

`onUpdateGameSession` (vedi [Inizializzare il 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 del codice al tuo client di gioco o al server di gioco per effettuare la chiamata [StopMatchmaking](#) all'GameLiftAPI Amazon. 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.

## Invia richieste di backfill (da un server di gioco)

Puoi avviare le richieste di backfill degli abbinamenti direttamente 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 attualmente connessi al gioco e sullo stato delle slot vuote.

Questo argomento presuppone che tu abbia già compilato i componenti di FlexMatch necessari e aggiunto correttamente i processi di abbinamento al server di gioco e a un servizio di gioco lato client. Per ulteriori dettagli sulla configurazione di FlexMatch, consulta la sezione [FlexMatchintegrazione con Amazon GameLift 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. Consultare [Aggiorna i dati delle partite sul server di gioco](#).

Come per altre funzionalità del server, un server di gioco utilizza Amazon GameLift Server SDK. 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. Aggiungere del codice per creare e inviare le richieste di backfill degli abbinamenti a un matchmaker FlexMatch. Le richieste di backfill vengono gestite tramite queste API server:

- [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.

- Ticket ID: fornisci un ID del ticket per il matchmaking (o scegli di farli generare automaticamente). È possibile utilizzare lo stesso meccanismo per assegnare gli ID dei ticket alle richieste di backfill e di abbinamento. Tali ticket vengono elaborati nello stesso modo.
- Matchmaker: identifica quale matchmaker utilizzare per la richiesta di backfill. 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 sono memorizzate nell'oggetto della sessione di gioco ([GameSession](#)), che è stato fornito al processo del server da Amazon al GameLift momento dell'attivazione della 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 \(giocatore\)](#) relative a tutti i giocatori in corso nella sessione di gioco che stai compilando. 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 su quali giocatori sono attualmente connessi alla sessione di gioco.
  2. Per ottenere gli ID dei giocatori, gli attributi e le assegnazioni delle squadre, estrai i dati dei giocatori dall'oggetto della sessione di gioco ([GameSession](#)), dalla `MatchmakerData` proprietà (vedi [Lavora con i 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 backfill. Amazon GameLift aggiorna il tuo server di gioco sullo stato delle richieste di backfill utilizzando la funzione di callback `Server SDK onUpdateGameSession` (vedi [Inizializzare il processo del server](#)). Aggiungi codice per gestire i messaggi di stato, nonché gli oggetti della sessione di gioco aggiornati a seguito di richieste di riempimento 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.

## Inviare richieste di riempimento (da un servizio clienti)

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 compilato i componenti di FlexMatch necessari e aggiunto correttamente i processi di abbinamento al server di gioco e a un servizio di gioco lato client.



Per ulteriori dettagli sulla configurazione di FlexMatch, consulta la sezione [FlexMatchintegrazione con Amazon GameLift 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, consultare [Aggiorna i dati delle partite sul server di gioco](#).

Come per altre funzionalità client, un servizio di gioco lato client utilizza l'AWSSDK con l'API Amazon GameLift. Questo SDK è disponibile in C++, C# e altri linguaggi. Per una descrizione generale delle API dei client, consulta Amazon GameLift Service API Reference, che descrive l'API di servizio di basso livello per le azioni GameLift relative ad Amazon e include collegamenti a guide di riferimento specifiche per la 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 di 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 GameLift servizio Amazon 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. Aggiungere del codice per creare e inviare le richieste di backfill degli abbinamenti a un matchmaker FlexMatch. Le richieste di backfill vengono gestite tramite queste API del client:

- [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.

- Ticket ID: fornisci un ID del ticket per il matchmaking (o scegli di farli generare automaticamente). È possibile utilizzare lo stesso meccanismo per assegnare gli ID dei ticket alle richieste di backfill e di abbinamento. 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 una `MatchmakerData` proprietà dell'oggetto della sessione di gioco ([GameSession](#)), sotto la 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 nuove richieste di matchmaking non ottengono un ID di sessione di gioco finché la partita non viene piazzata; la presenza dell'ID della sessione di gioco è un modo per distinguere tra i biglietti per le nuove partite e i biglietti per i backfill.
- Dati del giocatore: includi le informazioni sul [giocatore \(giocatore\)](#) relative a tutti i giocatori in corso nella sessione di gioco che stai compilando. 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 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 [Lavora con i dati del matchmaker](#)). Utilizzare gli ID giocatore acquisiti nella fase precedente per ottenere i dati soltanto sui 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 backfill delle partite nel tuo gioco, il tuo server di gioco deve essere in grado di gestire gli aggiornamenti delle sessioni di gioco che Amazon GameLift fornisce in seguito alle richieste di backfill delle partite.

Quando Amazon GameLift completa una richiesta di match backfill, con successo o meno, chiama il tuo server di gioco utilizzando la funzione di callback `onUpdateGameSession`. Questa chiamata ha tre parametri di input: un ID del ticket di backfill della partita, un messaggio di stato e un `GameSession` oggetto contenente la maggior parte dei dati di up-to-date matchmaking, comprese le informazioni sul giocatore. 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_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 backfill della corrispondenza 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 una corrispondenza di riempimento entro il limite di tempo. L'oggetto `GameSession` non viene modificato.
  - `BACKFILL_CANCELLED` — La richiesta di backfill della partita è stata annullata da una chiamata a `StopMatchmaking` (`client`) o (`server`). `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 tuo server di gioco all'azione Server SDK [ProcessReady\(\)](#), aggiungi il nome del metodo di `onUpdateGameSession` callback come parametro di processo.

# GameLiftFlexMatchRiferimento Amazon

Questa sezione contiene la documentazione di riferimento per il matchmaking con Amazon GameLiftFlexMatch.

## Argomenti

- [Riferimento GameLift FlexMatch API Amazon \(AWSSDK\)](#)
- [FlexMatchlinguaggio delle regole](#)
- [FlexMatcheventi di matchmaking](#)

## Riferimento GameLift FlexMatch API Amazon (AWSSDK)

Questo argomento fornisce un elenco di operazioni API per Amazon basato sulle attività. GameLift FlexMatch L'API del GameLift FlexMatch servizio Amazon è inclusa nell'AWSSDK nel namespace. `aws.gamelift` [Scarica l'AWSSDK](#) o [visualizza la documentazione di riferimento dell'GameLiftAPI Amazon](#).

Amazon GameLift FlexMatch fornisce servizi di matchmaking da utilizzare con giochi ospitati con soluzioni di GameLift hosting Amazon (incluso hosting gestito per server di gioco personalizzati o server in tempo reale e hosting su Amazon EC2 con Amazon GameLift FleetIQ), nonché con altri sistemi di hosting come primitive on-premise o peer-to-peer cloud computing. Consulta la [Amazon GameLift Developer Guide](#) per ulteriori informazioni su altre opzioni GameLift di hosting Amazon.

## Imposta regole e processi di matchmaking

Richiama queste operazioni per creare un FlexMatch matchmaker, configurare il processo di matchmaking per il tuo gioco e definire una serie di regole personalizzate per creare 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 utilizzi Amazon GameLift per l'hosting, specifica anche come creare una nuova sessione di gioco per la partita.
- [DescribeMatchmakingConfigurations](#)— Recupera le configurazioni di matchmaking definite in una regione Amazon. GameLift

- [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 di partite tra giocatori.
- [DescribeMatchmakingRuleSets](#)— Recupera i set di regole di matchmaking definiti in una regione AmazonGameLift.
- [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 vuole giocare nella stessa partita.
- [DescribeMatchmaking](#)— Ottieni dettagli su una richiesta di matchmaking, incluso lo stato.
- [AcceptMatch](#)— Per una partita che richiede l'accettazione da parte del giocatore, invia una notifica ad Amazon GameLift quando un giocatore accetta una partita proposta.
- [StopMatchmaking](#)— Annulla una richiesta di matchmaking.
- [StartMatchBackfill](#)— Richiedi partite aggiuntive per riempire gli spazi vuoti in una sessione di gioco esistente.

## Linguaggi di programmazione compatibili

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

- [C++ \(documenti SDK\) \(Amazon\) GameLift](#)
- [Java \(documenti SDK\) \(Amazon\) GameLift](#)
- [.NET \(documenti SDK\) \(Amazon\) GameLift](#)

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

## FlexMatchlinguaggio 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 FlexMatch Per assistenza dettagliata sulla scrittura di regole e set di regole di matchmaking, vedere [Crea un set di FlexMatch regole](#).

### Argomenti

- [FlexMatchschema del set di regole](#)
- [FlexMatchdefinizioni delle proprietà dei set di regole](#)
- [FlexMatchtipi di regole](#)
- [FlexMatchespressioni di proprietà](#)

## FlexMatchschema del set di regole

FlexMatch i set di regole utilizzano lo schema standard per le regole per abbinamenti piccoli e abbinamenti grandi. Per una descrizione dettagliata di ogni sezione, vedere [FlexMatchdefinizioni delle proprietà dei set di regole](#).

### Schema del set di regole per piccole partite

Lo schema seguente documenta tutte le proprietà possibili 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">,
    "sortAttribute": "string",
    "mapKey": <"minValue", "maxValue">,
    "partyAggregation": <"avg", "min", "max">
  },{
    "type": "absoluteSort",
    "name": "string",
    "description": "string",
    "sortDirection": <"ascending", "descending">,
    "sortAttribute": "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 proprietà possibili e i valori consentiti per un set di regole utilizzato per creare partite con più di 40 giocatori. Se il totale dei `maxPlayers` valori di tutte le squadre del set di regole è superiore a 40, FlexMatch elabora le richieste di partite che utilizzano questo set di regole secondo le linee guida relative alle partite ampie.

```
{
  "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  
    }  
  ]  
}
```

## FlexMatchdefinizioni delle proprietà dei set di regole

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

### **name**

Un'etichetta descrittiva per il set di regole. Questo valore non è associato al nome assegnato alla GameLift [MatchmakingRuleSet](#) risorsa Amazon. Questo valore è incluso nei dati di matchmaking che descrivono una partita completata, ma non viene utilizzato da alcun processo AmazonGameLift.

Valori consentiti: String

Obbligatorio? No

### **ruleLanguageVersion**

La versione del linguaggio delle espressioni delle FlexMatch proprietà utilizzato.

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 del giocatore nei dati di matchmaking che vengono passati 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 quando si costruiscono fiammiferi. Se questa proprietà non è impostata, il comportamento predefinito è «ExhaustiveSearch».

Valori consentiti:

- «ExhaustiveSearch»: metodo di abbinamento standard. FlexMatch forma una corrispondenza in base al ticket più vecchio di un batch valutando gli altri ticket del pool in base a una serie di regole di abbinamento personalizzate. Questa strategia viene utilizzata per partite con 40 giocatori o meno. Quando si utilizza questa strategia, `batchingPreference` deve essere impostato su «casuale» o «ordinato».

- «bilanciato»: metodo ottimizzato per formare rapidamente partite di grandi dimensioni. Questa strategia viene utilizzata solo per partite da 41 a 200 giocatori. Organizza le 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 medi di abilità di tutte le squadre in una partita. Quando si utilizza questa strategia, `balancedAttribute` deve essere impostato e `batchingPreference` deve essere impostato 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 ticket per il match building. Il preordinamento del pool di biglietti fa sì che i biglietti vengano raggruppati in base a una caratteristica specifica, il che tende ad aumentare l'uniformità tra i giocatori nelle partite finali.

Valori consentiti:

- «random» — Valido solo con `strategy = «ExhaustiveSearch»`. Non viene effettuato alcun pre-ordinamento; i biglietti nel pool vengono raggruppati casualmente. 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 dei giocatori elencati in `sortByAttributes`.
- «LargestPopulation»: valido solo con `strategy = «balanced»`. Il pool di ticket è preordinato per regioni in cui i giocatori segnalano livelli di latenza accettabili. Questo è il comportamento predefinito per una strategia bilanciata.
- «fastestRegion»: valido solo con `strategy = «balanced»`. Il pool di ticket 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 a essere bassa.

Obbligatorio? Sì

## **balancedAttribute**

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

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

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

### **sortByAttributes**

Un elenco di attributi dei giocatori da utilizzare per preordinare il pool di ticket prima della creazione in batch. Questa proprietà viene utilizzata solo quando si preordina 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 prioritizzazione per abbinare i ticket di riempimento. Questa proprietà determina quando FlexMatch elabora i ticket di backfill in un batch. Viene utilizzato solo quando si preordina 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 new match) non viene preso in considerazione quando si formano le partite.
- «alto»: un batch di ticket viene ordinato per tipo di richiesta (e quindi per età) e FlexMatch tenta innanzitutto di abbinare i ticket di riempimento.
- «basso»: un batch di ticket viene ordinato per tipo di richiesta (e quindi per età) e FlexMatch tenta innanzitutto di abbinare i ticket non di backfill.

Obbligatorio? No

### **expansionAgeSelection**

Il metodo per calcolare il tempo di attesa per l'espansione di una regola di corrispondenza. Le espansioni vengono utilizzate per ridurre i requisiti di una partita se una partita non è stata 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 nella partita parzialmente completata. Le espansioni

tendono ad essere attivate più lentamente, perché un ticket più recente può riavviare l'orologio del tempo di attesa.

- «più vecchio»: il tempo di attesa per l'espansione viene calcolato in base al ticket con il timestamp di creazione più vecchio 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 la squadra. È possibile fare riferimento ai nomi delle squadre nelle regole e nelle espansioni. In una partita di successo, i giocatori vengono assegnati in base al nome della squadra nei dati del 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 della partita è valido.

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 sono indicati 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 sull'uso di ogni tipo di regola, vedere [FlexMatch tipi di regole](#).

Valori consentiti:

- «absoluteSort»: ordina utilizzando un metodo di ordinamento esplicito che ordina i biglietti in un batch in base al confronto tra un attributo del giocatore specificato e il ticket più vecchio del batch.
- «collezione»: valuta i valori di una collezione, ad esempio un attributo del giocatore che è una collezione o un insieme di valori per più giocatori.
- «confronto»: confronta due valori.



- «compound»: definisce una regola di matchmaking composta utilizzando una combinazione logica di altre regole nel set di regole. Supportato solo per partite con 40 o meno giocatori.
- «distanza»: misura la distanza tra i valori numerici.
- «BatchDistance»: misura la differenza tra il valore di un attributo e lo utilizza per raggruppare le richieste di corrispondenza.
- «DistanceSort»: ordina utilizzando un metodo di ordinamento esplicito che ordina i biglietti in un batch in base al confronto tra un attributo giocatore specificato con un valore numerico e il ticket più vecchio del batch.
- «latenza»: valuta i dati di latenza regionali riportati per una richiesta di matchmaking.

Obbligatorio? Sì

## expansions

Regole per allentare i requisiti di una partita nel tempo quando una partita non può essere completata. Imposta le espansioni come una serie di passaggi da applicare gradualmente per rendere più facile trovare le corrispondenze. Per impostazione predefinita, FlexMatch calcola il tempo di attesa in base all'età dell'ultimo biglietto aggiunto a una partita. È possibile modificare la modalità di calcolo dei tempi di attesa dell'espansione utilizzando la proprietà dell'algorithmo `expansionAgeSelection`.

I tempi di attesa per l'espansione sono valori assoluti, quindi ogni passaggio dovrebbe avere un tempo di attesa più lungo del 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 partita, impostato nella configurazione del matchmaking.

Obbligatorio? No

## target

L'elemento del set di regole per rilassarsi. È possibile ridurre le proprietà relative alle dimensioni del team o qualsiasi proprietà delle istruzioni delle regole. La sintassi è "`<component name>[<rule/team name>]. <property name>`". Per 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.

## FlexMatchtipi di regole

### Regola della distanza tra batch

```
batchDistance
```

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

- **Confronta i valori numerici degli attributi.** Ad esempio, una regola di distanza tra gruppi 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 da FlexMatch utilizzare per formare i batch.

### Proprietà delle regole di distanza in batch

- **batchAttribute**— Il valore dell'attributo del giocatore utilizzato per formare i batch.
- **maxDistance**— Il valore massimo della distanza per una partita di successo. Utilizzato per confrontare gli attributi numerici.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (party). Le opzioni valide includono i valori minimo (`minmax`), massimo (`()`) e medio (`avg`) per i giocatori di un biglietto. 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, specificate 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 equal (=) o not-equal (!=). Non specificare un valore di riferimento.

### Note

Le regole sulla 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 possibile corrispondenza.
- **operation**— Il valore che determina come confrontare la misurazione con il valore di riferimento. Le operazioni valide includono: <=&, =, !=, >, >=.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (party). Le opzioni valide includono i valori minimo (minmax), massimo () e medio (avg) per i giocatori di un biglietto. Il valore predefinito è avg.

## Regola della distanza

distance

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

### Note

Le regole sulla 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 misurare la distanza. Deve essere un attributo con un valore numerico.
- **referenceValue**— Il valore numerico per misurare la distanza rispetto a una partita potenziale.
- **minDistance/maxDistance**— Il valore minimo o massimo della distanza per una partita di successo.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (party). Le opzioni valide includono i valori minimo (minmax), massimo () e medio (avg) per i giocatori di un biglietto. Il valore predefinito è avg.

## Regola di raccolta

collection

Le regole di raccolta confrontano i valori di un gruppo di attributi dei giocatori con quelli di altri giocatori del batch 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 ne abbia almeno uno di un certo carattere.

### 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 possibile 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, vedere [Esempio 4: utilizza l'ordinamento esplicito per trovare gli abbinamenti migliori](#).
  - **contains**— Questa operazione misura il numero di raccolte di attributi del giocatore che contengono il valore di riferimento specificato. Per un esempio di regola che utilizza l'operazione contains, vedere [Esempio 3: imposta i requisiti a livello di team e i limiti di latenza](#).
  - **reference\_intersection\_count**— Questa operazione misura il numero di oggetti in una collezione di attributi del giocatore che corrispondono agli elementi della collezione di valori di riferimento. Puoi usare questa operazione per confrontare diversi attributi dei giocatori. Per un esempio di regola che confronta le raccolte di attributi di più giocatori, vedi. [Esempio 5: trovare le intersezioni degli attributi di più giocatori](#)
- **minCount/maxCount**— Il valore di conteggio minimo o massimo per una partita di successo.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (party). Per questo valore, puoi usare `union` per combinare gli attributi dei giocatori di tutti i giocatori del gruppo. Oppure puoi usare `intersection` gli attributi del giocatore che il gruppo ha in comune. Il valore predefinito è `union`.

## Regola composta

compound

Le regole composte utilizzano dichiarazioni logiche per formare partite di 40 o meno giocatori. 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 puoi espandere una regola composta utilizzando [regole di espansione](#), ma puoi 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 puoi usare 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 l'esito dell'argomentazione 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 posizione affinché la regola della latenza lo accetti. È possibile utilizzare questo tipo di regola con corrispondenze di grandi dimensioni specificando la `maxLatency` proprietà.

### Proprietà della regola di latenza

- **maxLatency**— Il valore di latenza massimo accettabile per una posizione. Se un ticket non ha località 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 biglietti che si trovano entro la distanza massima dal valore di riferimento della distanza danno come risultato una partita. Le opzioni valide includono i valori di latenza minima (`minavg`) e media (`()`) del giocatore.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (`party`). Le opzioni valide includono i valori minimo (`minmax`), massimo (`()`) e medio (`avg`) per i giocatori di un biglietto. 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 sulle politiche di latenza per le code, consulta [Creare una politica di latenza per un giocatore](#).

## Regola di ordinamento assoluto

```
absoluteSort
```

Le regole di ordinamento assolute 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` ed `descending`.
- **sortAttribute**— L'attributo del giocatore in base al quale ordinare i biglietti.
- **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 viene prima.
  - `maxValue`— La chiave con il valore più alto viene prima.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (`party`). Le opzioni valide includono l'attributo `minimum (min) player`, l'attributo `maximum (max) player` 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 per 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 della 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 biglietto aggiunto al batch.

### Proprietà della regola di ordinamento della distanza

- **sortDirection**— La direzione per ordinare i ticket di matchmaking. Le opzioni valide includono `ascending` ed `descending`.



- **sortAttribute**— L'attributo del giocatore in base al quale ordinare i biglietti.
- **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 ticket aggiunto al batch, trova la chiave con il valore più alto.
- **partyAggregation**— Il valore che determina la modalità FlexMatch di gestione dei ticket con più giocatori (party). Le opzioni valide includono i valori minimo (**minmax**), massimo (**max**) e medio (**avg**) per i giocatori di un biglietto. Il valore predefinito è **avg**.

## FlexMatchespressioni 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>

Obiettivo	Input	Significato	Output
Per identificare tutti i team in un abbinamento:	<code>teams[*]</code>	Tutti i team	<code>List&lt;Team&gt;</code>
Per identificare i giocatori in un determinato team:	<code>team[red].players</code>	Giocatori del team "red"	<code>List&lt;Player&gt;</code>
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	<code>List&lt;List&lt;Player&gt;&gt;</code>
Per identificare i giocatori in un abbinamento:	<code>team[*].players</code>	Giocatori nell'abbinamento, raggruppati per team	<code>List&lt;List&lt;Player&gt;&gt;</code>

## Esempi di espressioni di proprietà

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

Espressione	Significato	Tipo risultante
<code>teams[red].players[playerId]</code>	ID giocatore di tutti i giocatori del team "red"	<code>List&lt;string&gt;</code>
<code>teams[red].players.attributes[skill]</code>	Attributi "skill" di tutti i giocatori del team "red"	<code>List&lt;number&gt;</code>
<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	<code>List&lt;List&lt;number&gt;&gt;</code>

Espressione	Significato	Tipo risultante
<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
<code>min</code>	List<number>	Ottiene il minimo di tutti i numeri nell'elenco.	numero
<code>max</code>	List<number>	Ottiene il massimo di tutti i numeri nell'elenco.	numero
<code>avg</code>	List<number>	Ottiene la media di tutti i numeri nell'elenco.	numero
<code>median</code>	List<number>	Ottiene la mediana di tutti i numeri nell'elenco.	numero
<code>sum</code>	List<number>	Ottiene la somma di tutti i numeri nell'elenco.	numero
<code>count</code>	List<?>	Ottiene il numero di elementi nell'elenco.	numero

Aggregazione	Input	Significato	Output
<code>stddev</code>	<code>List&lt;number&gt;</code>	Ottiene la deviazione e standard di tutti i numeri nell'elenco.	numero
<code>flatten</code>	<code>List&lt;List&lt;?&gt;&gt;</code>	Trasforma una raccolta di elenchi nidificati in un singolo elenco contenente tutti gli elementi.	<code>List&lt;?&gt;</code>
<code>set_intersection</code>	<code>List&lt;List&lt;string&gt;&gt;</code>	Ottiene un elenco di stringhe che si trovano in tutti gli elenchi di stringhe di una raccolta.	<code>List&lt;string&gt;</code>
Tutte le operazioni	<code>List&lt;List&lt;?&gt;&gt;</code>	Tutte le operazioni su un elenco nidificato o operano singolarmente su ogni sottoelenco per ottenere un elenco di risultati.	<code>List&lt;?&gt;</code>

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

Espressione	Significato	Tipo risultante
<code>flatten(teams[*].players.attributes[skill])</code>	Attributi "skill" di tutti i giocatori dell'abbonamento (non raggruppati)	<code>List&lt;number&gt;</code>

Espressione	Significato	Tipo risultante
<code>avg(teams[red].players.attributes[skill])</code>	Livello di competenza a medio dei giocatori del team "red"	numero
<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.	numero
<code>count(teams[red].players)</code>	Numero di giocatori del team "red"	numero
<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	numero

## FlexMatcheventi di matchmaking

Amazon GameLift 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 [Configurare le notifiche FlexMatch degli eventi](#). Questi eventi vengono inoltre trasmessi ad Amazon CloudWatch Events quasi in tempo reale e con il massimo impegno possibile.

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

## 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

Dettaglio: tipologia, bigliettiestimatedWaitMillis, 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, biglietti, tempistica di accettazione, accettazione richiesta,, 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 AcceptMatch non è stato richiesto per quel giocatore.

Risorsa: ConfigurationArn

Dettaglio: 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

Dettaglio: tipologia, 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

Dettagli: tipo, ticket, messaggioruleEvaluationMetrics, 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

Dettagli: tipo, ticket, messaggioruleEvaluationMetrics, 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

Dettagli: tipo, ticket, messaggioruleEvaluationMetrics, 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"
}
```

# Protezione con FlexMatch

Per AWS, la sicurezza del cloud ha 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 AWS e l'utente. Per informazioni su come applicare il modello di responsabilità condivisa durante l'utilizzo FlexMatch, consulta [Sicurezza in Amazon GameLift](#).

# Note GameLift FlexMatch di rilascio di Amazon e versioni SDK

Le note di GameLift rilascio di Amazon forniscono dettagli su nuove FlexMatch funzionalità, aggiornamenti e correzioni relativi al servizio. Questa pagina include anche la cronologia delle versioni di Amazon GameLift SDK.

# Risorse per GameLift sviluppatori Amazon

Per visualizzare tutta GameLift la documentazione Amazon e le risorse per gli sviluppatori, consulta la home page di [Amazon GameLift Documentation](#).

# Glossario per AWS

Per la terminologia AWS più recente, consultare il [glossario AWS](#) nella documentazione di riferimento per Glossario AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.