



Guia do Desenvolvedor

Amazon GameLift



Amazon GameLift: Guia do Desenvolvedor

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

Table of Contents

O que é o Amazon GameLift?	1
Usos do Amazon GameLift	1
Começar a usar soluções do Amazon GameLift	1
Hospedagem do Amazon GameLift para servidores personalizados	2
Hospedagem do Amazon GameLift com os servidores em tempo real	2
Amazon GameLift FleetIQ para hospedagem no Amazon EC2	3
Amazon GameLift FlexMatch para marcação de jogos	4
Hospedagem de hardware Anywhere do Amazon GameLift	4
Acessar o Amazon GameLift	4
Preços do Amazon GameLift	5
Como o Amazon GameLift funciona	5
Componentes principais	6
Hospedar servidores de jogos	6
Execução de sessões de jogos	7
Escalar a capacidade de frota	8
Monitoramento do Amazon GameLift	8
Usando outros recursos da AWS	9
Como os jogadores se conectam aos jogos	9
Arquitetura de jogos com Amazon gerenciada GameLift	10
Configuração	13
Configurar uma conta	13
Inscreva-se para um Conta da AWS	14
Criar um usuário com acesso administrativo	14
Gerenciar permissões de usuário para a Amazon GameLift	15
Configurar o acesso programático, consulte para usuários	16
Configurar o acesso programático para o jogo	18
Exemplo de permissões do IAM	19
Configurar um perfil de serviço do IAM	23
Suporte ao desenvolvimento	26
Para servidores de jogos personalizados	27
Para serviços personalizados ao cliente	29
Para servidores em tempo real	29
Gerencie os custos de hospedagem de jogos	30
Crie alertas de cobrança para monitorar o uso	30

Rastreie os custos por GameLift frota da Amazon	30
Defina a capacidade não utilizada da frota como zero	31
Locais GameLift de hospedagem da Amazon	31
GameLift Hospedagem Amazon	31
Zonas Locais	33
Amazon GameLift Anywhere	34
Amazon GameLift FlexMatch	34
Amazon GameLift na China	35
Conceitos básicos	36
Exemplo de servidor de jogo personalizado	36
Exemplo de jogo de servidores em tempo real	36
Roteiro de hospedagem gerenciada	38
Escolha uma opção de hospedagem	38
Preparar seu jogo	40
Prepare o servidor de jogos personalizado	40
Preparar seu servidor em tempo real	41
Teste a integração	41
Planeje e implante seus recursos	42
Implantar seus recursos	43
Projete seu serviço de back-end	43
Autenticar os jogadores	44
Back-end de tecnologia sem servidor	44
Back-end baseado em WebSocket	46
Configurar métricas e log	48
Listas de verificação de lançamento	49
Integração	49
Testar	50
Executar	51
Pós-lançamento	51
Preparando jogos para a Amazon GameLift	52
Integre jogos com servidores de jogos personalizados	52
Interações do Amazon GameLift	53
Integrar um servidor de jogos	58
Integrar um cliente de jogo	68
Mecanismos de jogo e Amazon GameLift	75
Teste sua integração (SDK 5 do servidor)	102

Teste sua integração (SDK 4 do servidor)	109
Integrar jogos com servidores em tempo real	118
O que são servidores em tempo real?	118
Gerenciar sessões de jogo	119
Interação entre servidores e clientes	119
Personalizar um servidor	120
Implantação e atualização	121
Integrar um cliente do jogo	121
Personalizar um script em tempo real	127
Integrando jogos com o plug-in para Unity	133
Guia do plug-in para Unity (SDK 5.x do servidor)	133
Guia do plug-in para Unity (SDK do servidor 4.x)	152
Integrando jogos com o plugin para Unreal	179
Sobre o plug-in	179
Fluxo do plug-in	180
Instale o plug-in para Unreal	181
Configurar um perfil de usuário da AWS	184
Configure seu jogo com Anywhere	186
Implante seu jogo com frotas gerenciadas do Amazon EC2	200
Obter dados de frotas	204
Adicionar a marcação de jogos do FlexMatch	205
Gerenciando hospedagem com contêineres [Pré-visualização]	206
Atributos principais	206
Usando frotas de contêineres durante a pré-visualização pública	207
Como os contêineres funcionam	207
componentes da frota de contêineres	207
Arquiteturas comuns	210
Conceitos principais	211
Roteiro de desenvolvimento	215
Integre seu jogo com a Amazon GameLift	217
Ferramentas de integração	218
Crie seu servidor de jogos para Linux	219
Teste a integração com uma frota Anywhere	219
Preparar uma imagem de contêiner	221
Crie um diretório de trabalho	222
Crie sua imagem	223

Empurre sua imagem	232
Projete uma frota de contêineres	233
Arquitete sua estrutura de contêineres de frota	233
Defina limites de recursos	235
Designe recipientes essenciais	237
Configurar conexões de rede	237
Configurar verificações de saúde para contêineres	241
Definir dependências de contêineres	242
Configurar uma frota de contêineres	242
Crie definições de grupos de contêineres	244
Antes de começar	245
Clonar uma definição de grupo de contêineres	245
Criar uma definição de grupo de contêineres de réplica	246
Criar um JSON arquivo de definição de contêiner	249
Crie uma frota de contêineres	251
Gerencie suas frotas de contêineres	257
Visualizar recursos da	257
Atualizar recursos	257
Excluir recursos	258
Dimensionamento de frotas de contêineres	258
Gerenciar recursos de hospedagem	261
Fazer upload de compilações e scripts	262
Fazer upload de uma compilação	263
Carregar um script	272
Configurar frotas	277
Guia de criação de frota	278
Criar um nova frota	286
Gerencie suas frotas	303
Adicionar um alias a uma frota	306
Depurar problemas de frota	308
Conecte-se remotamente às instâncias da frota	312
Escalar a capacidade de hospedagem	320
Para gerenciar a capacidade da frota no console	321
Definir limites de capacidade de hospedagem	321
Definir manualmente capacidade de frota	323
Capacidade de frota de ajuste de escala automático	326

Configurar filas	333
Criar uma fila	333
Práticas recomendadas	342
Criar uma fila	343
Configurar notificações de eventos	347
Tutorial: Filas para instâncias spot	351
Gerenciar recursos com o AWS CloudFormation	359
Práticas recomendadas	360
Usar pilhas do AWS CloudFormation	361
Atualizar compilações	365
emparelhamento de VPC	368
Como configurar o emparelhamento de VPC para uma frota existente	368
Como configurar o emparelhamento de VPC com uma nova frota	371
Solucionar problemas de emparelhamento de VPC	374
Visualizar dados do jogo	376
Veja seu GameLift status na Amazon	376
Exibir as compilações	378
Detalhes da compilação	379
Visualizar seus scripts	379
Detalhes do script	380
Visualizar as frotas	380
Visualizar detalhes da frota	381
Detalhes	381
Métricas	382
Eventos	382
Escalabilidade	383
Localizações	384
Sessões de jogos	384
Visualizar informações das sessões de jogo e jogador	384
Detalhes	385
Sessões de jogador	386
Informações do jogador	386
Visualizar os aliases	386
Detalhes do alias	387
Visualizar as filas	388
Visualizar detalhes da fila	388

Monitorando a Amazon GameLift	391
Monitorar com o CloudWatch	392
Dimensões de métricas	392
Métricas de frota	393
Métricas de fila	406
Métricas do FlexMatch	409
Métricas do FleetIQ	413
Registrar em log chamadas de API	416
Informações sobre o Amazon GameLift no CloudTrail	416
Noções básicas sobre entradas do arquivo de log do Amazon GameLift	417
Registrando mensagens ao servidor	419
Registro em log para servidores personalizados	420
Registro para servidores em tempo real	422
Segurança	428
Proteção de dados	429
Criptografia inativa	431
Criptografia em trânsito	431
Privacidade do tráfego entre redes	431
Gerenciamento de identidade e acesso	432
Público	432
Autenticando com identidades	433
Gerenciamento do acesso usando políticas	437
Como a Amazon GameLift trabalha com o IAM	439
Exemplos de políticas baseadas em identidade	448
Solução de problemas	453
Registro em log e monitoramento com o Amazon GameLift	455
Validação de conformidade	456
Resiliência	457
Segurança da infraestrutura	459
Análise de configuração e vulnerabilidade	459
Melhores práticas de segurança	460
Não abra portas para a Internet	460
Saiba mais	461
Guias de referência do Amazon GameLift	462
Referência de API do serviço (SDK da AWS)	462
Configurar e gerenciar os recursos de hospedagem do Amazon GameLift	462

Inicie sessões de jogo e junte-se aos jogadores	466
Referência dos servidores em tempo real	468
Referência de API (C#) do cliente em tempo real	468
Referência de script de servidores em tempo real	483
Referência do SDK de servidor	491
Referência do SDK do servidor para C++	491
Referência do SDK do servidor para C#	568
Referência do SDK do servidor para Go	633
Referência de Server SDK para Unreal Engine	661
Eventos de posicionamento de sessões de jogo	723
Sintaxe do evento de posicionamento	723
PlacementFulfilled	724
PlacementCancelled	726
PlacementTimedOut	727
PlacementFailed	728
Fazer a estimativa do preço	730
Fazer a estimativa da hospedagem do Amazon GameLift	730
Instâncias do Amazon GameLift	730
DTO (transferência de dados de saída)	733
Estime o FlexMatch autônomo do Amazon GameLift	733
Cotas e regiões suportadas	736
Notas de lançamento e versões do SDK	737
Versões do SDK	737
Notas de release	743
Glossário do AWS	775
.....	dcclxxvi

O que é o Amazon GameLift?

Use o Amazon GameLift para implantar, operar e escalar servidores dedicados e de baixo custo na nuvem para jogos multijogador baseados em sessão. Compilado na infraestrutura global de computação da AWS, o Amazon GameLift ajuda a entregar servidores de jogos de alto desempenho, alta confiabilidade enquanto dimensiona o uso de recursos de forma dinâmica para atender á demanda mundial de jogadores.

Usos do Amazon GameLift

O Amazon GameLift é compatível com estes casos de uso e muito mais:

- Utilize seus próprios servidores de jogos multijogador personalizados ou utilize os servidores em tempo real prontos para usar para hospedar seus jogos.
- Execute recursos de hospedagem de baixo custo usando [instâncias spot do Amazon Elastic Compute Cloud \(Amazon EC2\)](#).
- Dimensione automaticamente a quantidade de recursos de hospedagem que seu jogo precisa com base no uso.
- Gerencie os recursos de computação do Amazon EC2 em um só lugar usando o Amazon GameLift FleetIQ.
- Combine jogadores em jogos multijogador com o Amazon GameLift FlexMatch.
- Teste iterativamente suas compilações de servidor e cliente de jogos com o Anywhere do Amazon GameLift.
- Use seu próprio hardware enquanto gerencia tudo em um só lugar com o Anywhere do Amazon GameLift.

Tip

Para experimentar a hospedagem do servidor de jogos Amazon GameLift, consulte [Conceitos básicos do Amazon GameLift](#).

Começar a usar soluções do Amazon GameLift

Soluções Amazon GameLift para desenvolvedores de jogos

- [Hospedagem do Amazon GameLift para servidores personalizados](#)
- [Hospedagem do Amazon GameLift com os servidores em tempo real](#)
- [Amazon GameLift FleetIQ para hospedagem no Amazon EC2](#)
- [Amazon GameLift FlexMatch para marcação de jogos](#)
- [Hospedagem de hardware Anywhere do Amazon GameLift](#)

Hospedagem do Amazon GameLift para servidores personalizados

O Amazon GameLift substitui o trabalho necessário para hospedar seus próprios servidores de jogos personalizados. Os recursos de ajuste de escala automático ajudam você a evitar pagar por mais recursos do que o necessário. O ajuste de escala automático também ajuda a garantir que você sempre tenha jogos disponíveis para novos jogadores entrarem com o mínimo de espera.

Para obter mais informações sobre hospedagem do Amazon GameLift, consulte [Como o Amazon GameLift funciona](#).

Recursos principais

- Use os recursos de gerenciamento do Amazon GameLift, inclusive o ajuste de escala automático, filas em vários locais e colocação de sessões de jogos.
- Implante servidores de jogos a serem executados nos sistemas operacionais Amazon Linux ou Windows Server.
- Gerencie sessões de jogo e sessões de jogador.
- Configure o acompanhamento de integridade personalizado para processos de servidor a fim de detectar problemas e resolver processos com mau desempenho.
- Gerencie os recursos de jogo usando modelos do AWS CloudFormation para o Amazon GameLift.

Hospedagem do Amazon GameLift com os servidores em tempo real

Use os servidores em tempo real para compilar jogos que não precisam de servidores de jogos personalizados. Esta solução de servidor leve fornece servidores de jogos que você poderá configurar para se adequar ao jogo

Para obter mais informações sobre a hospedagem do Amazon GameLift com servidores em tempo real, consulte [Integração de jogos com Servidores em Tempo Real do Amazon GameLift](#).

Recursos principais

- Use os recursos de gerenciamento do Amazon GameLift, inclusive o ajuste de escala automático, filas em vários locais e colocação de sessões de jogos.
- Use os recursos de hospedagem do Amazon GameLift e escolha o tipo de hardware de computação da AWS para suas frotas.
- Aproveite uma pilha de rede completa para interação com o cliente e servidor de jogos.
- Obtenha a funcionalidade principal do servidor de jogos com lógica de servidor personalizável.
- Faça atualizações em tempo real das configurações do Realtime e da lógica de servidor.

Amazon GameLift FleetIQ para hospedagem no Amazon EC2

Use o Amazon GameLift FleetIQ para trabalhar diretamente com seus recursos de hospedagem no Amazon EC2 e no Amazon EC2 Auto Scaling. Isso oferece o benefício das otimizações do Amazon GameLift para hospedagem de jogos econômica e resiliente. Essa solução é para desenvolvedores de jogos que precisam de mais flexibilidade do que as soluções totalmente gerenciadas do Amazon GameLift oferecem.

Para obter informações sobre como o Amazon GameLift FleetIQ funciona com o Amazon EC2 e o EC2 Auto Scaling para hospedagem de jogos, consulte o [Guia do desenvolvedor do Amazon GameLift FleetIQ](#).

Recursos principais

- Obtenha balanceamento otimizado de instâncias spot usando o algoritmo FleetIQ.
- Use os recursos de roteamento de jogadores para gerenciar os recursos do servidor de jogos com eficiência e fornecer uma melhor experiência ao jogador ao participar de jogos.
- Escale automaticamente a capacidade de hospedagem com base no uso do jogador.
- Gerencie as instâncias do Amazon EC2 diretamente em sua própria Conta da AWS.
- Use qualquer um dos formatos executáveis de servidor de jogo compatíveis, incluindo Windows, Linux, contêineres e Kubernetes.

Amazon GameLift FlexMatch para marcação de jogos

Use o FlexMatch para criar conjuntos de regras personalizados para definir partidas multijogador para o jogo. O FlexMatch usa conjuntos de regras para comparar jogadores compatíveis em cada partida e fornecer aos jogadores a experiência multijogador ideal.

Para obter mais informações sobre o FlexMatch, consulte [O que é o Amazon GameLift FlexMatch?](#)

Recursos principais

- Equilibre a velocidade de criação da partida e a qualidade da partida.
- Combine jogadores ou equipes com base em características definidas.
- Defina regras para colocar os jogadores nas partidas com base na latência.

Hospedagem de hardware Anywhere do Amazon GameLift

Use o Anywhere do Amazon GameLift para integrar hardware em qualquer lugar do seu ambiente à sua hospedagem de jogos do Amazon GameLift. Integre frotas Anywhere e frotas EC2 em filas de marcador de jogos e de sessões de jogos para gerenciar a marcação de jogos e posicionamento de jogos em seu hardware.

Para obter mais informações sobre testes com o Anywhere consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#). Para obter mais informações sobre a configuração de uma frota Anywhere, consulte [Configurar as frotas do Amazon GameLift](#).

Recursos principais

- Execute testes rápidos e iterativos das versões do servidor e do cliente do jogo.
- Use as ferramentas definidas do Amazon GameLift para implantar jogos em seu próprio hardware.
- Use o hardware mais próximo de seus players, em qualquer lugar.

Acessar o Amazon GameLift

Use essas ferramentas para trabalhar com o Amazon GameLift.

SDKs do Amazon GameLift

Os SDKs do Amazon GameLift contêm as bibliotecas necessárias para se comunicar com o Amazon GameLift a partir de clientes de jogos, servidores de jogos e serviços de jogos. Para obter mais informações, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).

SDK do cliente em tempo real do Amazon GameLift

O SDK do cliente em tempo real permite que um cliente de jogo se conecte ao servidor em tempo real, participe de sessões de jogo e permaneça sincronizado com outros jogadores. Faça download do [SDK](#) e saiba mais sobre como fazer chamadas da API com o [API cliente dos servidores em tempo real \(C#\)](#).

Console do Amazon GameLift

Use o [AWS Management Console do Amazon GameLift](#) para gerenciar suas implantações de jogos, configurar recursos e rastrear métricas de uso e desempenho do jogador. O console do Amazon GameLift fornece uma GUI alternativa para gerenciar recursos de maneira programática com o AWS Command Line Interface (AWS CLI).

AWS CLI

Use esta ferramenta da linha de comando para fazer chamadas ao SDK da AWS, inclusive à API do Amazon GameLift. Para obter informações sobre como usar a AWS CLI, consulte [Introdução à AWS CLI](#) no Guia do usuário da AWS Command Line Interface.

Preços do Amazon GameLift

O Amazon GameLift cobra pelas instâncias por duração de uso e pela largura de banda por quantidade de dados transferidos. Para obter uma lista completa de cobranças e preços do Amazon GameLift, consulte [Definição de preço do Amazon GameLift](#).

Para obter informações sobre como calcular o custo de hospedar seus jogos ou marcação de jogos com o Amazon GameLift, veja [Gerando estimativas de preços do Amazon GameLift](#), que descreve como usar o [AWS Pricing Calculator](#).

Como o Amazon GameLift funciona

Este tópico abrange os principais componentes para hospedagem de jogos e descreve como o Amazon GameLift disponibiliza seus servidores de jogos multijogador para os jogadores.

Pronto para preparar seu jogo para hospedagem no Amazon GameLift? Confira o [Roteiro de hospedagem gerenciada do Amazon GameLift](#).

Componentes principais

Configurar o Amazon GameLift para hospedar o jogo envolve trabalhar com os seguintes componentes. O diagrama em [Arquitetura de jogos com Amazon gerenciada GameLift](#) visualiza as relações entre esses componentes.

- Um servidor de jogos é o software de servidor do seu jogo executado em uma frota. Você carrega a compilação ou o script do seu servidor de jogo para o Amazon GameLift e informa ao Amazon GameLift. Ao usar o Anywhere do Amazon GameLift ou o Amazon GameLift FleetIQ, você carrega a compilação do servidor de jogos diretamente no recurso de computação.
- Uma sessão de jogo é um jogo em andamento com jogadores. Defina as características básicas de uma sessão de jogos, como duração e quantidade de jogadores. Os jogadores então se conectam ao servidor do jogo para participar de uma sessão de jogo.
- Um cliente do jogo é o software do jogo em execução no dispositivo de um jogador. Um cliente de jogo se conecta a um servidor de jogos por meio de serviços de back-end para participar de uma sessão de jogo, com base nas informações de conexão que ele recebe do Amazon GameLift.
- Os serviços de back-end são serviços adicionais e personalizados que lidam com tarefas relacionadas ao Amazon GameLift. Como melhor prática, seus serviços de back-end devem lidar com toda a comunicação do cliente do jogo com o Amazon GameLift.

Hospedar servidores de jogos

Com o Amazon GameLift, hospede seus servidores de jogos de três maneiras diferentes: Amazon GameLift gerenciado, Amazon GameLift FleetIQ e Anywhere do Amazon GameLift. Para obter mais informações sobre o Amazon GameLift FleetIQ, consulte [O que é o Amazon GameLift FleetIQ?](#)

Você pode projetar uma frota de acordo com as necessidades do seu jogo. Para obter mais informações sobre como projetar uma frota, consulte [Guia de criação de frota do Amazon GameLift](#).

Amazon GameLift gerenciado

Com o Amazon GameLift gerenciado, hospede seus servidores de jogos nos recursos de computação virtual do Amazon GameLift, chamados de instâncias. Configure seus recursos de hospedagem criando uma frota de instâncias e implantando-as para executar os servidores de jogos.

Anywhere do Amazon GameLift

Com o Anywhere Amazon GameLift, hospede seus servidores de jogos na computação que você gerencia. Configure seus recursos de hospedagem criando uma frota Anywhere que faça referência à sua computação.

Aliases de frota

Um alias é uma designação que você poderá transferir entre frotas, tornando-se uma maneira conveniente de ter uma localização genérica da frota. Use um alias para alternar as frotas usadas pelos clientes do jogo sem precisar alterar o cliente do jogo. Também é possível criar um alias de terminal que você aponta para o conteúdo.

Execução de sessões de jogos

Depois de implantar sua compilação de servidor de jogos em uma frota e o Amazon GameLift iniciar processos de servidor de jogos em cada instância, a frota poderá hospedar sessões de jogo. O Amazon GameLift inicia novas sessões de jogo quando seu serviço de cliente de jogos envia uma solicitação de posicionamento para o serviço de back-end ou para o Amazon GameLift.

Posicionamento da sessão de jogo e o algoritmo FleetIQ

As filas usam o algoritmo FleetIQ para selecionar um servidor de jogos disponível para hospedar uma nova sessão de jogo. O principal componente para a colocação de sessões de jogo é a fila de sessões de jogos do Amazon GameLift. Você atribui a uma fila de sessões de jogo uma lista de frotas, que determina onde a fila pode colocar as sessões de jogo. Para obter mais informações sobre filas de sessões de jogo e como criá-las para seu jogo, consulte [Criar uma fila de sessões de jogo](#).

Conexões de jogadores com jogos

Como parte do processo de colocação da sessão de jogo, a fila ou sessão de jogo solicita que o servidor de jogo selecionado inicie uma nova sessão de jogo. O servidor do jogo responde à solicitação e reporta ao Amazon GameLift quando estiver pronto para aceitar conexões de jogadores. Em seguida, o Amazon GameLift entrega as informações de conexão ao serviço de back-end ou ao serviço de cliente do jogo. Seus clientes de jogo usam essas informações para se conectarem diretamente à sessão de jogo e iniciar o jogo.

Escalar a capacidade de frota

Quando a frota estiver ativa e pronta para hospedar sessões de jogos, poderá ajustar a capacidade da frota para atender à demanda dos jogadores. É recomendável buscar um equilíbrio entre todos os novos jogadores encontrarem um jogo rapidamente e gastarem demais em recursos que ficam ociosos.

O Amazon GameLift fornece uma ferramenta de ajuste de escala automático altamente eficaz, ou definia manualmente a capacidade da frota. Para obter mais informações, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#).

Ajuste de escala automático

O Amazon GameLift fornece dois métodos de ajuste de escala automático:

- [Ajuste de escala automático baseado em destino](#)
- [Ajuste de escala automático com políticas com base nas regras](#)

Recursos de escalabilidade adicionais

- Proteção de sessão de jogo – Impede que o Amazon GameLift encerre sessões de jogo que hospedam jogadores ativos durante um evento de redução de escala.
- Limites de escalabilidade – Controla uso geral de instâncias definindo limites mínimos e máximos para o número de instâncias em uma frota.
- Suspensão do ajuste de escala automático – Suspenda o ajuste de escala automático no nível da localização da frota sem alterar ou excluir suas políticas de ajuste de escala automático.
- Métricas de escalabilidade – Rastreie o histórico de capacidade e eventos de escalabilidade de uma frota.

Monitoramento do Amazon GameLift

Quando você tem frotas ativas e em execução, o Amazon GameLift coleta uma variedade de informações para ajudar a monitorar o desempenho dos servidores de jogos implantados. Use essas informações para otimizar o uso de recursos, resolver problemas e obter informações sobre como os jogadores permanecem ativos nos jogos. O Amazon GameLift coleta o seguinte:

- Detalhes da frota, localização, sessão de jogo e sessão do jogador

- Métricas de uso
- Integridade do processo do servidor
- Logs de sessão de jogo

Para obter mais informações sobre monitoramento no Amazon GameLift, consulte [Monitorando a Amazon GameLift](#).

Usando outros recursos da AWS

Seus servidores e aplicativos de jogos podem se comunicar com outros recursos da AWS. Por exemplo, use um conjunto de serviços da Web para autenticação de jogador ou redes sociais. Para que seus servidores de jogos acessem os recursos da AWS que você gerencia sua Conta da AWS, permita explicitamente que o Amazon GameLift acesse seus recursos AWS.

O Amazon GameLift fornece algumas opções de gerenciamento desse tipo de acesso. Para obter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Como os jogadores se conectam aos jogos

Uma sessão de jogo é uma instância do jogo em execução no Amazon GameLift. Para jogar o jogo, um jogador pode encontrar e ingressar em uma sessão de jogo existente ou criar uma sessão de jogo nova e adicioná-la. Um jogador ingressa criando uma sessão de jogador para a sessão de jogo. Se a sessão do jogo estiver aberta para jogadores, o Amazon GameLift reserva um espaço para o jogador e fornecerá informações de conexão. O jogador pode acabar se conectando à sessão de jogo e reivindicando o slot reservado.

Para obter informações detalhadas sobre como criar e gerenciar sessões de jogos e sessões de jogadores com servidores de jogos personalizados, consulte [Adicione GameLift a Amazon ao seu cliente de jogos](#). Para obter informações sobre como conectar jogadores a servidores em tempo real, consulte [Integrar um cliente do jogo para servidores em tempo real](#).

O Amazon GameLift fornece vários recursos relacionados às sessões de jogo e jogador.

Hospedar sessões de jogo nos melhores recursos disponíveis entre vários locais

Escolha uma dentre várias opções ao configurar como o Amazon GameLift seleciona recursos para hospedar novas sessões de jogo. Caso esteja administrando frotas em vários locais,

poderá criar filas de sessões de jogo que coloquem novas sessões de jogo em qualquer frota, independentemente da localização.

Controlar acesso do jogador a sessões de jogos

Configure sessões de jogo para permitir ou negar solicitações de entrada de novos jogadores, independentemente do número de jogadores conectados.

Usar dados de jogo e jogador personalizados

Adicione dados personalizados aos objetos da sessão do jogo e aos objetos da sessão do jogador. O Amazon GameLift passa dados da sessão de jogo para um servidor de jogos ao início de uma nova sessão de jogo. O Amazon GameLift passa os dados da sessão do jogador para o servidor do jogo quando um jogador se conecta à sessão do jogo.

Filtrar e classificar sessões de jogos disponíveis

Use a pesquisa e classificação da sessão para encontrar a melhor correspondência possível para um jogador em potencial ou permita o jogador escolher em uma lista de sessões de jogo disponíveis. Use a pesquisa e classificação de sessões para encontrar sessões de jogo com base nas características da sessão. Também é possível pesquisar e classificar de acordo com seus próprios dados personalizados do jogo.

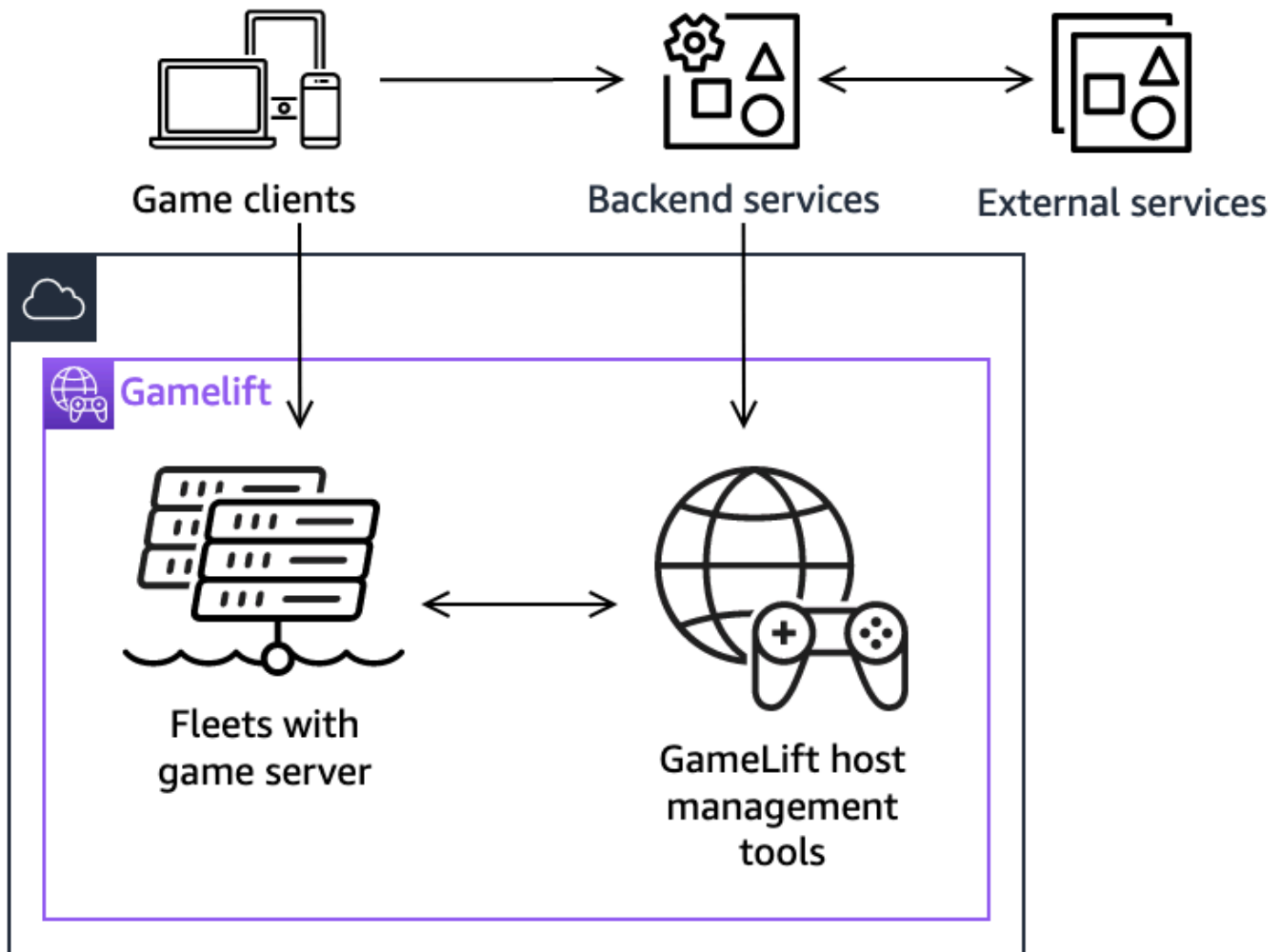
Acompanhar dados de uso de jogo e jogador

Armazene automaticamente registros de sessões de jogo concluídas. Configure o armazenamento em log ao integrar o Amazon GameLift aos servidores de jogos. Para obter mais informações, consulte [Registro de mensagens do servidor no Amazon GameLift](#).

Use o console do Amazon GameLift para visualizar informações detalhadas sobre sessões de jogo, inclusive metadados de sessão, configurações e dados de sessão do jogador. Para obter mais informações, consulte [Visualizar dados nas sessões de jogo e jogador](#) e [Métricas](#).

Arquitetura de jogos com Amazon gerenciada GameLift

O diagrama a seguir ilustra os principais componentes de uma arquitetura de jogo hospedada usando a GameLift solução gerenciada da Amazon.



Os principais componentes desta arquitetura incluem o seguinte:

Cientes do jogo

Para participar de um jogo hospedado na Amazon GameLift, seu cliente de jogo deve primeiro encontrar uma sessão de jogo disponível. O cliente do jogo pesquisa sessões de jogo existentes, solicita matchmaking ou inicia uma nova sessão de jogo comunicando-se com a Amazon por GameLift meio de um serviço de back-end. O serviço de back-end faz solicitações à Amazon e GameLift, em resposta, recebe informações da sessão do jogo, que são retransmitidas ao cliente do jogo. O cliente do jogo então se conecta ao servidor do jogo. Para ter mais informações, consulte [Preparando jogos para a Amazon GameLift](#).

Serviços de back-end

Um serviço de back-end gerencia a comunicação entre clientes de jogos e a Amazon GameLift chamando as operações GameLift de API do serviço Amazon no AWS SDK. Também é possível usar serviços de back-end para outras tarefas específicas do jogo, como autenticação e autorização do jogador, inventário ou controle de moeda. Para ter mais informações, consulte [Projete seu serviço de cliente de jogos](#).

Serviços externos

O jogo pode depender de um serviço externo, como para validar uma associação por assinatura. Um serviço externo pode passar informações para seus servidores de jogos por meio de um serviço de back-end e da Amazon GameLift.

Servidores de jogos

Você carrega seu software de servidor de jogos para a Amazon GameLift, e a Amazon GameLift o implanta em máquinas de hospedagem para hospedar sessões de jogo e aceitar conexões de jogadores. Os servidores de jogos se comunicam com GameLift a Amazon para iniciar sessões de jogo, validar jogadores recém-conectados e relatar o status das sessões de jogo, conexões de jogadores e recursos disponíveis.

Servidores de jogos personalizados se comunicam com a Amazon GameLift usando o Amazon GameLift Server SDK. Clientes de jogos se conectam diretamente a um servidor de jogos após receberem detalhes de conexão da Amazon GameLift por meio de um serviço de back-end. Para ter mais informações, consulte [Integre jogos com servidores de jogos personalizados](#).

Servidores em tempo real são servidores de jogos que executam seu script personalizado. Ao entrar em um jogo, um cliente de jogo se conecta diretamente a um servidor em tempo real usando o SDK do cliente em tempo real. Para ter mais informações, consulte [Integração de jogos com Servidores em Tempo Real do Amazon GameLift](#).

Ferramentas de gerenciamento de host

Ao configurar e gerenciar recursos de hospedagem, os proprietários de jogos usam ferramentas de gerenciamento de hospedagem para gerenciar compilações ou scripts de servidores de jogos, frotas, matchmaking e filas. A GameLift ferramenta Amazon definida no AWS SDK e no console fornece várias maneiras de gerenciar seus recursos de hospedagem. Acesse remotamente qualquer servidor de jogo individual para solução de problemas.

Configuração

Obtenha ajuda com a configuração de sua Conta da AWS para usar o Amazon GameLift e hospedar seus jogos multijogador.

Tip

Para experimentar a hospedagem do servidor de jogos Amazon GameLift, consulte [Conceitos básicos do Amazon GameLift](#).

Tópicos

- [Configurar um Conta da AWS](#)
- [Suporte ao desenvolvimento com a Amazon GameLift](#)
- [Gerencie os custos de hospedagem de jogos](#)
- [Locais GameLift de hospedagem da Amazon](#)

Configurar um Conta da AWS

Para começar a usar a Amazon GameLift, crie e configure seu Conta da AWS. Não há custo para criar uma Conta da AWS. Esta seção explica como criar sua conta, configurar seus usuários e configurar permissões.

Tópicos

- [Inscreva-se para um Conta da AWS](#)
- [Criar um usuário com acesso administrativo](#)
- [Gerenciar permissões de usuário para a Amazon GameLift](#)
- [Configurar o acesso programático, consulte para usuários](#)
- [Configurar o acesso programático para o jogo](#)
- [Exemplos de permissões do IAM do Amazon GameLift](#)
- [Configurar uma função de serviço do IAM para a Amazon GameLift](#)

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário raiz, consulte [Signing in as the root user](#) (Fazer login como usuário raiz) no Guia do usuário Início de Sessão da AWS .

2. Ative a autenticação multifator (MFA) para seu usuário raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário IAM Identity Center, use a URL de login enviada ao seu endereço de e-mail quando você criou o usuário IAM Identity Center user.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

Gerenciar permissões de usuário para a Amazon GameLift

Crie usuários adicionais ou estenda as permissões de acesso aos usuários existentes conforme necessário para seus GameLift recursos da Amazon. Como prática recomendada ([Melhores práticas de segurança no IAM](#)), aplique permissões de privilégio mínimo para todos os usuários. Para obter orientação sobre a sintaxe de permissões, consulte [Exemplos de permissões do IAM do Amazon GameLift](#).

Use as instruções a seguir para definir permissões de usuário com base em como você gerencia os usuários em sua AWS conta.

Para conceder acesso, adicione as permissões aos seus usuários, grupos ou perfis:

- Usuários e grupos em AWS IAM Identity Center:

Crie um conjunto de permissões. Siga as instruções em [Criação de um conjunto de permissões](#) no Guia do usuário do AWS IAM Identity Center .

- Usuários gerenciados no IAM com provedor de identidades:

Crie um perfil para a federação de identidades. Siga as instruções em [Criar um perfil para um provedor de identidades de terceiros \(federação\)](#) no Guia do usuário do IAM.

- Usuários do IAM:

- Crie um perfil que seu usuário possa assumir. Siga as instruções em [Criação de um perfil para um usuário do IAM](#) no Guia do usuário do IAM.

- (Não recomendado) Vincule uma política diretamente a um usuário ou adicione um usuário a um grupo de usuários. Siga as instruções em [Adição de permissões a um usuário \(console\)](#) no Guia do usuário do IAM.

Ao trabalhar com usuários do IAM, como melhor prática, sempre atribua permissões a funções ou grupos de usuários e não a usuários individuais.

Configurar o acesso programático, consulte para usuários

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do AWS Management Console. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
Identificação da força de trabalho	Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
(Usuários gerenciados no Centro de Identidade do IAM)		<ul style="list-style-type: none">• Para o AWS CLI, consulte Configurando o AWS CLI para uso AWS IAM Identity Center no Guia do AWS Command Line Interface usuário.• Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para AWS SDKs ou APIs. AWS CLI AWS	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para AWS SDKs AWS CLI ou APIs. AWS	Siga as instruções da interface que deseja utilizar. <ul style="list-style-type: none"> • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de AWS SDKs e ferramentas. • Para AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

Se você usa chaves de acesso, consulte [Práticas recomendadas para gerenciar chaves de AWS acesso](#).

Configurar o acesso programático para o jogo

A maioria dos jogos usa serviços de back-end para se comunicar com a Amazon GameLift usando os AWS SDKs. Por exemplo, você usa um serviço de back-end (agindo em nome de clientes de jogos) para solicitar sessões de jogo, colocar jogadores em jogos e outras tarefas. Esses serviços precisam de acesso programático e credenciais de segurança para autenticar chamadas para as APIs de serviços da Amazon GameLift .

Para a Amazon GameLift, você gerencia esse acesso criando um usuário player no AWS Identity and Access Management (IAM). Gerencie as permissões do usuário jogador por meio de uma das seguintes opções:

- Crie um perfil do IAM com permissões de usuário jogador e permita que o usuário jogador assuma o perfil quando necessário. O serviço de back-end deve incluir código para assumir essa função antes de fazer solicitações à Amazon GameLift. De acordo com as melhores práticas de segurança, os perfis fornecem acesso limitado e temporário. Você pode usar funções para cargas de trabalho executadas em AWS recursos ([funções do IAM](#)) ou fora deles AWS ([funções do IAM em qualquer lugar](#)).
- Crie um grupo de usuários do IAM com permissões de usuário do jogador e adicione seu usuário jogador ao grupo. Essa opção fornece credenciais de usuário de longo prazo ao jogador, que o serviço de back-end deve armazenar e usar ao se comunicar com a Amazon. GameLift

Para obter a sintaxe da política de permissões, consulte [Exemplos de permissão de usuário do jogador](#).

Para obter mais informações sobre o gerenciamento de permissões para uso por um workload, consulte [Identidades do IAM: credenciais temporárias no IAM](#).

Exemplos de permissões do IAM do Amazon GameLift

Use a sintaxe nesses exemplos para definir permissões AWS Identity and Access Management (IAM) para usuários que precisam acessar os recursos do Amazon GameLift. Para obter mais informações sobre como gerenciar permissões de usuário, consulte [Gerenciar permissões de usuário para a Amazon GameLift](#). Ao gerenciar permissões para usuários fora do IAM Identity Center, como melhor prática, sempre atribua permissões ao perfil do IAM ou grupos de usuários, não a usuários individuais.

Caso esteja usando o Amazon GameLift FleetIQ como uma solução independente, consulte [Configurar sua Conta da AWS para o Amazon GameLift FleetIQ](#).

Exemplos de permissões de administrador

Esses exemplos dão ao usuário acesso total para gerenciar os recursos de hospedagem de jogos do Amazon GameLift.

Exemplo Sintaxe das permissões de recursos do Amazon GameLift

O exemplo a seguir estende o acesso a todos os recursos do Amazon GameLift.

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

Example Sintaxe para permissões de recursos do Amazon GameLift com suporte para regiões que não estão habilitadas por padrão

O exemplo a seguir estende o acesso a todos os recursos da AWS e regiões do Amazon GameLift que não estão habilitados por padrão. Para obter mais informações sobre regiões que não estão habilitadas por padrão e como habilitá-las, consulte [Gerenciar as Regiões da AWS](#) no Referência geral da AWS.

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

Example Sintaxe para recursos e permissões **PassRole** do Amazon GameLift

O exemplo a seguir estende o acesso a todos os recursos do Amazon GameLift e permite que um usuário transmita um perfil de serviço do IAM ao Amazon GameLift. Uma função de serviço dá ao Amazon GameLift capacidade limitada de acessar outros recursos e serviços em seu nome, conforme descrito em [Configurar uma função de serviço do IAM para a Amazon GameLift](#). Por exemplo, ao responder a uma `CreateBuild` solicitação, o Amazon GameLift precisa acessar seus arquivos de compilação em um bucket do Amazon S3. Para obter mais informações sobre a ação de `PassRole`, consulte [IAM: passar um perfil de serviço para um serviço específico da AWS](#) no Guia do usuário do IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "gamelift:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "gamelift.amazonaws.com"
        }
      }
    }
  ]
}
```

Exemplos de permissão de usuário do jogador

Esses exemplos permitem que um serviço de back-end ou outra entidade faça chamadas de API para a API do Amazon GameLift. Eles abrangem os cenários comuns para gerenciar sessões de jogo, sessões de jogadores e marcação de jogos. Para obter mais detalhes, consulte [Configurar o acesso programático para o jogo](#).

Exemplo Sintaxe das permissões de colocação de sessão de jogo

O exemplo a seguir estende o acesso às APIs do Amazon GameLift que usam filas de posicionamento de sessões de jogo para criar sessões de jogo e gerenciar sessões de jogadores.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "PlayerPermissionsForGameSessionPlacements",
    "Effect": "Allow",
    "Action": [
      "gamelift:StartGameSessionPlacement",
      "gamelift:DescribeGameSessionPlacement",
      "gamelift:StopGameSessionPlacement",
```

```

    "gamelift:CreatePlayerSession",
    "gamelift:CreatePlayerSessions",
    "gamelift:DescribeGameSessions"
  ],
  "Resource": "*"
}
}

```

Example Sintaxe para permissões de marcação de jogos

O exemplo a seguir estende o acesso às APIs do Amazon GameLift que gerenciam as atividades de marcação de jogos do FlexMatch. O FlexMatch combina jogadores para sessões de jogo novas ou existentes e inicia a colocação de sessões de jogo para jogos hospedados no Amazon GameLift. Para obter mais informações sobre o FlexMatch, consulte [O que é o Amazon GameLift FlexMatch?](#)

```

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

```

Example Sintaxe para permissões manuais de posicionamento de sessões de jogos

O exemplo a seguir estende o acesso às APIs do Amazon GameLift que criam manualmente sessões de jogo e sessões de jogadores em frotas específicas. Esse cenário oferece suporte a jogos que não usam filas de posicionamento, como jogos que permitem que os jogadores participem escolhendo em uma lista de sessões de jogo disponíveis (o método “listar e escolher”).

```

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

```

```
"Statement": {
  "Sid": "PlayerPermissionsForManualGameSessions",
  "Effect": "Allow",
  "Action": [
    "gamelift:CreateGameSession",
    "gamelift:DescribeGameSessions",
    "gamelift:SearchGameSessions",
    "gamelift:CreatePlayerSession",
    "gamelift:CreatePlayerSessions",
    "gamelift:DescribePlayerSessions"
  ],
  "Resource": "*"
}
```

Configurar uma função de serviço do IAM para a Amazon GameLift

Alguns GameLift recursos da Amazon exigem que você estenda o acesso limitado aos AWS recursos que você possui. Para fazer isso, crie um perfil do AWS Identity and Access Management (IAM). Um [perfil do IAM](#) é uma identidade do IAM que você pode criar em sua conta que tem permissões específicas. Um perfil do IAM é semelhante a um usuário do IAM porque é uma identidade da AWS com políticas de permissão que determinam o que ela pode e não pode fazer na AWS. No entanto, em vez de ser exclusivamente associada a uma pessoa, o propósito do perfil é ser assumido por qualquer pessoa que precisar dele. Além disso, um perfil não tem credenciais de longo prazo padrão associadas a ele, como senha ou chaves de acesso. Em vez disso, quando você assumir um perfil, ele fornecerá credenciais de segurança temporárias para sua sessão de perfil.

Este tópico aborda como criar uma função que você pode usar com suas frotas GameLift gerenciadas pela Amazon. [Se você usa o Amazon GameLift FleetIQ para otimizar a hospedagem de jogos em suas instâncias do Amazon Elastic Compute Cloud \(Amazon EC2\), consulte Configurar seu perfil para o Amazon FleetIQ. Conta da AWS GameLift](#)

No procedimento a seguir, crie uma função com uma política de permissões personalizada e uma política de confiança que permita GameLift à Amazon assumir a função.


Criar um perfil do IAM personalizado

Etapa 1: criar uma política de permissões.

Para usar o editor de políticas JSON para criar uma política

1. Faça login no AWS Management Console e abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação à esquerda, escolha Políticas (Políticas).

Se essa for a primeira vez que você escolhe Políticas, a página Bem-vindo às políticas gerenciadas será exibida. Escolha Conceitos básicos.
3. Na parte superior da página, escolha Criar política.
4. Na seção Editor de políticas, escolha a opção JSON.
5. Insira ou cole um documento de política JSON. Para obter detalhes sobre a linguagem da política do IAM, consulte a referência de [política JSON do IAM](#).
6. Resolva os avisos de segurança, erros ou avisos gerais gerados durante a [validação de política](#) e depois escolha Avançar.

 Note

Você pode alternar entre as opções de editor Visual e JSON a qualquer momento. Porém, se você fizer alterações ou escolher Próximo no editor Visual, o IAM poderá reestruturar a política a fim de otimizá-la para o editor visual. Para obter mais informações, consulte [Reestruturação de política](#) no Guia do usuário do IAM.

7. (Opcional) Ao criar ou editar uma política no AWS Management Console, você pode gerar um modelo de política JSON ou YAML que pode ser usado em modelos do AWS CloudFormation.

Para fazer isso, no editor de políticas, escolha Ações e, em seguida, escolha Gerar CloudFormation modelo. Para saber mais sobre o AWS CloudFormation, consulte [Referência de tipos de recurso do AWS Identity and Access Management](#) no Guia do usuário do AWS CloudFormation.
8. Quando terminar de adicionar as permissões à política, escolha Avançar.
9. Na página Revisar e criar, insira um Nome de política e uma Descrição (opcional) para a política que você está criando. Revise Permissões definidas nessa política para ver as permissões que são concedidas pela política.
10. (Opcional) Adicione metadados à política associando tags como pares de chave-valor. Para obter mais informações sobre o uso de tags no IAM, consulte [Marcar recursos do IAM](#) no Guia do usuário do IAM.

11. Escolha Criar política para salvar sua nova política.

Etapa 2: Crie uma função que a Amazon GameLift possa assumir.

1. No painel de navegação do console do IAM, escolha Perfis e, em seguida, Criar perfil.
2. Na página Selecionar entidade confiável, escolha a opção Política de confiança personalizada. Esta seleção abre o editor de Política de confiança personalizada.
3. Substitua a sintaxe JSON padrão pela seguinte e, em seguida, escolha Avançar para continuar.

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

4. Na página Adicionar permissões, localize e selecione a política de permissões criada na Etapa 1. Escolha Avançar para continuar.
5. Na página Nome, revisar e criar, insira um Nome de perfil e uma Descrição (opcional) para o perfil que você está criando. Revise as Entidades confiáveis e as Permissões adicionadas.
6. Escolha Criar perfil para salvar seu novo perfil.

Sintaxe da política de permissões

- Permissões para GameLift a Amazon assumir a função de serviço

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

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

- Permissões para acessar regiões da AWS que não estejam habilitadas por padrão

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

Suporte ao desenvolvimento com a Amazon GameLift

GameLift A Amazon fornece um conjunto de SDKs que você pode usar com suas soluções gerenciadas de hospedagem de jogos. Use GameLift os SDKs da Amazon para adicionar a funcionalidade necessária aos servidores de jogos multijogador, clientes de jogos e serviços de jogos que precisam interagir com o serviço de GameLift hospedagem da Amazon.

Para obter as informações mais recentes sobre as versões do Amazon GameLift SDK e a compatibilidade do SDK, consulte [Notas GameLift de lançamento da Amazon](#)

Para servidores de jogos personalizados

Crie e implante servidores de jogos personalizados de 64 bits com o Amazon GameLift Server SDK. Os servidores de jogos integrados ao SDK do servidor e implantados para hospedagem podem se comunicar com o GameLift serviço da Amazon para iniciar e gerenciar sessões de jogos. Para obter informações sobre a integração do SDK do servidor, consulte os tópicos em [Preparando jogos para a Amazon GameLift](#).

Sistemas operacionais de desenvolvimento

- Windows
- Linux

Linguagens de programação compatíveis

A Amazon GameLift fornece o SDK do servidor para os seguintes idiomas. [Baixe os SDKs do servidor](#). Para obter informações detalhadas específicas da versão, consulte os arquivos Readme incluídos em cada pacote.

- C++ server SDK
 - [Referência do SDK](#)
 - [Integração do SDK](#)
- SDK do servidor C# (as versões podem suportar o.NET 4 e .NET 6)
 - [Referência do SDK](#)
 - [Integração do SDK](#)
- Go
 - [Referência do SDK](#)
 - [Integração do SDK](#)

Mecanismos de jogo com suporte

Use SDKs específicos de linguagem com qualquer mecanismo compatível com bibliotecas C++, C# ou Go. Além disso, a Amazon GameLift fornece esses plug-ins de mecanismo de jogo: [Baixe os GameLift plug-ins da Amazon](#)

- Unity

- O plug-in SDK do servidor C# para Unity é um plug-in leve com bibliotecas pré-construídas que você pode instalar usando o gerenciador de pacotes do Unity. Use esse plug-in com as seguintes versões do Unity: 2020.3 LTS, 2021.3 LTS e 2022.3 LTS para Windows e Mac OS. Ele suporta os perfis .NET Framework e .NET Standard do Unity, com o .NET Standard 2.1 e .NET 4.x.
 - [Integre o Amazon GameLift em um projeto do Unity](#)
- O plug-in autônomo para Unity 2021.3 LTS e 2022.3 LTS é um plug-in completo com as bibliotecas C# SDK criadas para Unity e elementos de GUI para configurar e implantar recursos da Amazon para hospedagem. GameLift
 - [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#)
 - [Referência do SDK do servidor do Amazon GameLift para C#](#)
- Unreal Engine
 - O plug-in SDK do servidor C++ para Unreal é um plug-in leve que consiste no código-fonte C++ Unreal que você pode criar em bibliotecas para uso com as versões 4, 5 e 5.1 do Unreal Engine.
 - [Integre a Amazon GameLift em um projeto Unreal Engine](#)
 - [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)
 - O plug-in autônomo para Unreal Engine 5.0, 5.1 e 5.2 é um plug-in completo com o C++ para bibliotecas e SDK do servidor Unreal. AWS O plug-in é instalado no editor Unreal, com elementos de interface do usuário e materiais de apoio para configurar e implantar recursos da GameLift Amazon para hospedagem.
 - [Integrando jogos com o GameLift plugin Amazon para Unreal Engine](#)
 - [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)

Sistemas operacionais do servidor de jogos

Use o Amazon GameLift Server SDK para criar servidores de jogos para serem executados nas seguintes plataformas:

- [Windows Server 2016](#)
- [Amazon Linux 2023](#)
- [Amazon Linux 2 \(AL2\)](#)
- [Windows Server 2012](#) (consulte [GameLift Perguntas frequentes da Amazon para Windows 2012](#))
- [Amazon Linux \(AL1\)](#) (consulte [GameLift Perguntas frequentes da Amazon para AL1](#))

Para serviços personalizados ao cliente

Crie serviços de cliente personalizados de 64 bits usando o AWS SDK com a GameLift API da Amazon. Esse SDK permite que os serviços ao cliente gerenciem sessões de jogos e unam jogadores a jogos hospedados na Amazon GameLift. Para começar, [baixe o AWS SDK](#). Para obter mais informações sobre o uso do SDK com a Amazon GameLift, consulte a [Referência de GameLift API da Amazon](#).

Para servidores em tempo real

Configure e implante servidores em tempo real para hospedar seus jogos multijogador. Para permitir que seus clientes de jogos se conectem aos servidores em tempo real, use o Amazon GameLift Realtime Client SDK. Os clientes de jogos usam esse SDK para trocar mensagens com um servidor em tempo real e com outros clientes de jogos que se conectam ao servidor. Para começar, [baixe o Amazon GameLift Realtime Client SDK](#). Para obter informações sobre a configuração, consulte [Integrar um cliente do jogo para servidores em tempo real](#).

Compatibilidade com o SDK

O SDK do cliente em tempo real contém origem para as seguintes linguagens:

- C# (.NET)

Ambientes de desenvolvimento

Compile o SDK a partir da origem conforme necessário para os seguintes sistemas operacionais de desenvolvimento e mecanismos de jogos compatíveis:

- Sistemas operacionais – Windows, Linux, Android, iOS.
- Mecanismos de jogos – Unity, mecanismos que oferecem suporte para bibliotecas C#

Sistemas operacionais do servidor de jogos

É possível implantar servidores em tempo real em recursos de hospedagem executados nas seguintes plataformas:

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

Gerencie os custos de hospedagem de jogos

Sua fatura da AWS reflete os custos de hospedagem de jogos. Visualize as cobranças estimadas do mês atual e as cobranças finais dos meses anteriores no console de faturamento em <https://console.aws.amazon.com/billing/>. Para obter mais informações sobre ferramentas e recursos para ajudá-lo a gerenciar seus custos da AWS, consulte o [Guia do Usuário da AWS Billing](#). Este guia pode ajudá-lo a analisar o consumo de recursos, estabelecer o uso futuro e determinar suas necessidades de escalabilidade.

Em particular, considere essas dicas para ajudá-lo a gerenciar o custo dos GameLift serviços da Amazon.

Crie alertas de cobrança para monitorar o uso

Configure um alerta de uso do nível AWS gratuito para notificá-lo quando seu uso estiver próximo ou excedendo os limites do nível gratuito para a Amazon GameLift e outros. Serviços da AWS Configure os alertas para agir com base nos seus níveis de uso. Por exemplo, defina automaticamente seu orçamento como zero quando atingir o limite do nível gratuito.

Você também pode definir alertas de CloudWatch cobrança da Amazon para receber notificações quando o uso atingir limites personalizados.

Para obter mais informações, consulte um destes tópicos no Guia do usuário do AWS Billing:

- [Como monitorar o uso do nível gratuito da AWS](#)
- [Preferências de alerta de cobrança](#)

Rastreie os custos por GameLift frota da Amazon

Use tags de alocação de AWS custos para organizar e monitorar seus custos de hospedagem de jogos com base nas frotas do Amazon GameLift Amazon EC2 e em outros recursos do EC2. Ao marcar suas frotas, individualmente ou por grupos, crie relatórios de alocação de custos que categorizam os custos com base na etiqueta atribuída. Use esse tipo de relatório para identificar como as frotas estão contribuindo para os custos de hospedagem. Também é possível usar tags para filtrar visualizações no AWS Cost Explorer.

Para obter mais informações, consulte estes tópicos:

- [Uso de tags de alocação de custos da AWS](#) no Guia do usuário do AWS Billing.

- [Analisando seus custos com Cost Explorer da AWS](#), Guia do usuário do AWS Cost Management

Defina a capacidade não utilizada da frota como zero

As frotas podem continuar incorrendo em custos mesmo quando não estão em uso para hospedar sessões de jogos. Para evitar cobranças desnecessárias, [reduza sua frota](#) a zero quando não estiver em uso. Caso use o ajuste de escala automático, suspenda essa atividade e defina manualmente a capacidade da frota.

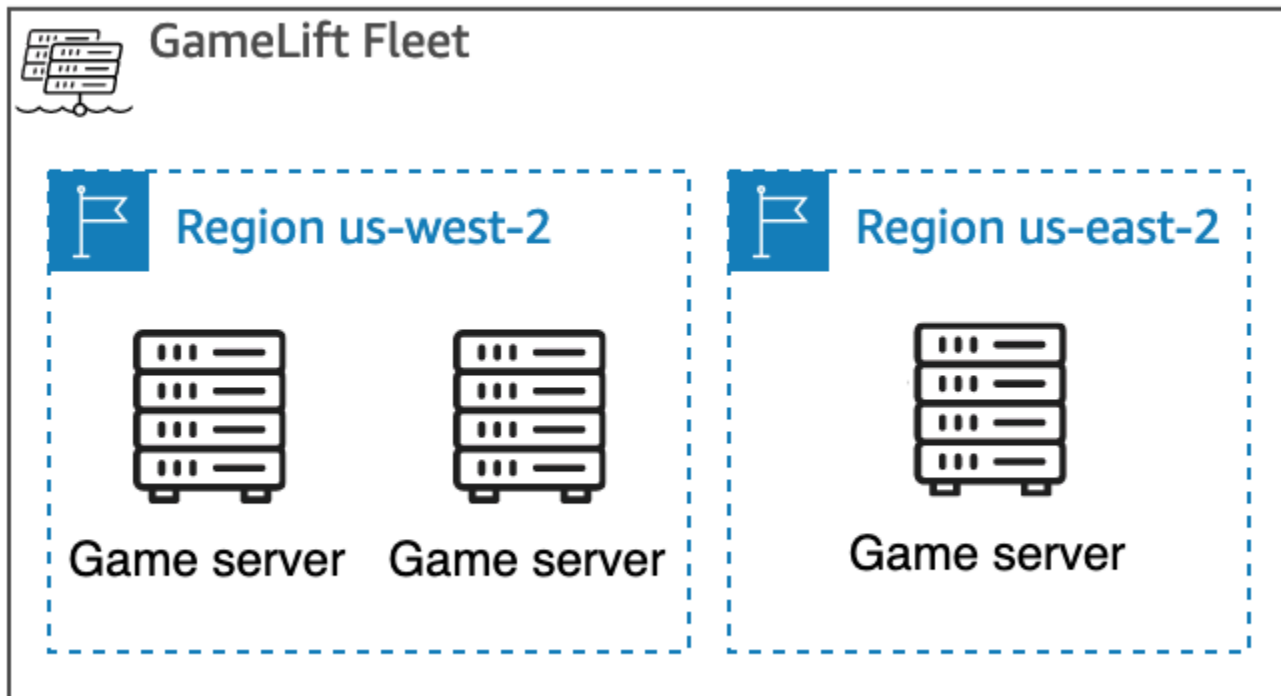
Locais GameLift de hospedagem da Amazon

A Amazon GameLift está disponível em várias Regiões da AWS Zonas Locais. Para obter uma lista completa de locais, consulte [GameLiftendpoints e cotas da Amazon](#) no. Referência geral da AWS

GameLift Hospedagem Amazon

Quando você cria uma GameLift frota da Amazon, a Amazon GameLift cria os recursos da frota em sua frota atualRegião da AWS. A Amazon GameLift chama essa região de região de origem da frota. Para gerenciar uma frota, acesse-a de sua região de origem.

Frotas de vários locais implantam instâncias em outros locais além da região de origem da frota. Com frotas com vários locais, você pode gerenciar a capacidade de cada local individualmente e organizar sessões de jogo por local. As frotas com vários locais podem ter localizações remotas em qualquer região ou zona local compatível com a Amazon GameLift . O diagrama a seguir mostra uma frota de vários locais com recursos em duas regiões. No diagrama, a região us-west-2 inclui dois servidores de jogos e a região us-east-2 tem um servidor de jogos.



Caso opte por usar uma [frota de vários locais](#) com instâncias em regiões que não estão habilitadas por padrão, habilite essas regiões na sua Conta da AWS. Além disso, sua política de GameLift administrador da Amazon deve permitir a `ec2:DescribeRegions` ação. Para obter mais informações sobre as regiões que não estão habilitadas por padrão e como habilitá-las, consulte [Gerenciar Regiões da AWS](#) no Referência geral da AWS. Para obter um exemplo de política com regiões que não estão habilitadas por padrão, consulte [Exemplos de permissões de administrador](#).

⚠ Important

Para usar regiões que não estão habilitadas por padrão, ative-as na sua Conta da AWS.

- As frotas com regiões não habilitadas que você criou antes de 28 de fevereiro de 2022 não são afetadas.
- Para criar novas frotas com vários locais ou atualizar frotas com vários locais existentes, primeiro ative todas as regiões que você escolher usar.

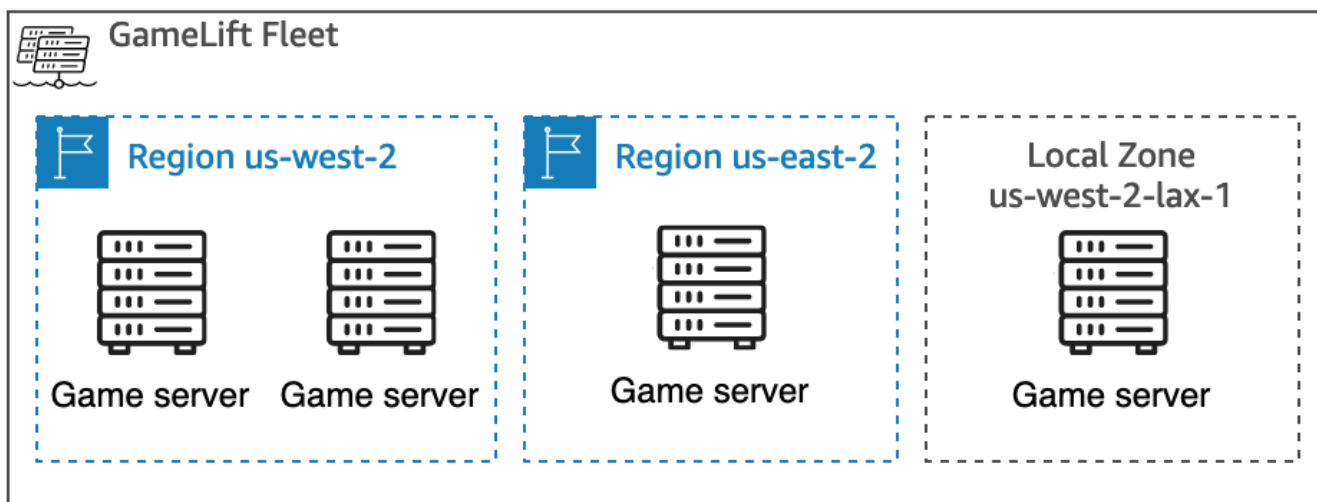
Para posicionamento de sessões de jogo, você pode criar filas de sessões de jogo em qualquer local GameLift compatível com a Amazon. A Amazon GameLift coloca sessões de jogo a partir do local em que você criou a fila.

Zonas Locais

Uma zona local é uma extensão de uma Região da AWS geograficamente próxima aos usuários. As zonas locais têm suas próprias conexões com a internet. As zonas locais também oferecem suporte ao AWS Direct Connect para que os recursos criados em uma zona local possam atender usuários locais com comunicações de baixa latência. Para obter mais informações, consulte [Zonas Locais da AWS](#).

O código da zona local é o código da região da seguido por um identificador que indica o local físico. Por exemplo, a zona local `us-west-2-lax-1` é em Los Angeles. Para obter uma lista das zonas locais disponíveis, consulte [Zonas locais disponíveis](#).

A Amazon GameLift hospeda seus jogos em cada um dos locais que você escolher para sua frota. O diagrama a seguir mostra uma frota com dois servidores de jogos na região `us-west-2`, um servidor de jogos na região `us-east-2` e um servidor de jogos na zona local `us-west-2-lax-1`.



Zonas locais disponíveis

A tabela a seguir lista as zonas locais disponíveis e suas localizações físicas.

Zona local	Localização (região metropolitana)
<code>us-east-1-atl-1</code>	Atlanta
<code>us-east-1-chi-1</code>	Chicago

Zona local	Localização (região metropolitana)
us-east-1-dfw-1	Dallas
us-east-1-iah-1	Houston
us-east-1-mci-1	Cidade de Kansas
us-west-2-den-1	Denver
us-west-2-lax-1	Los Angeles
us-west-2-phx-1	Phoenix

Amazon GameLift Anywhere

Você pode usar GameLift Anywhere a Amazon para criar frotas com seu próprio hardware e gerenciar suas compilações de jogos, scripts, servidores de jogos e clientes usando a Amazon. GameLift A Amazon GameLift Anywhere está disponível em todas as regiões que a Amazon GameLift oferece suporte. Para obter mais informações sobre como criar uma frota Anywhere e testar a integração do servidor de jogos, consulte [Crie uma GameLift Anywhere frota da Amazon](#) e [Teste sua integração usando frotas Anywhere do Amazon GameLift](#).

Com a Amazon, GameLift Anywhere você cria locais personalizados que representam a localização física do hardware que você está usando para hospedar seus servidores de jogos GameLift integrados da Amazon.

Amazon GameLift FlexMatch

Pois FlexMatch, você pode hospedar sessões de jogos geradas por partidas em qualquer local compatível com a Amazon GameLift. A atividade real de matchmaking ocorre Região da AWS onde você escolheu criar seus recursos de matchmaker. A Amazon GameLift encaminha as solicitações para o matchmaker e as processa nesse local. Para obter mais informações sobre a Amazon GameLift FlexMatch, consulte [O que é a Amazon GameLift FlexMatch?](#)

[Regiões da AWS que apoiam FlexMatch recursos](#)

Amazon GameLift na China

Ao usar a Amazon GameLift para obter recursos na região da China (Pequim), operada pela Sinnet, ou na região da China (Ningxia), operada pela NWCD, você deve ter uma conta separada AWS (China). Observe que alguns recursos não estão disponíveis nas regiões da China. Para obter mais informações sobre o uso da Amazon GameLift nessas regiões, consulte os seguintes recursos:

- [Amazon Web Services na China](#)
- [Amazon GameLift](#) (Introdução à Amazon Web Services na China)

Conceitos básicos do Amazon GameLift

Recomendamos que você experimente os exemplos a seguir antes de usar o Amazon GameLift em seu próprio jogo. O exemplo de servidor de jogo personalizado oferece experiência com hospedagem de jogos no console Amazon GameLift. O exemplo de servidores em tempo real mostra como preparar um jogo para hospedagem usando servidores em tempo real.

Para começar a usar o Amazon GameLift para seu próprio jogo, consulte [Roteiro de hospedagem gerenciada do Amazon GameLift](#).

Exemplo de servidor de jogo personalizado

Este exemplo demonstra um jogo personalizado dinâmico no Amazon GameLift. O exemplo conduz você pelas etapas a seguir:

- Criação de um exemplo de compilação de jogo.
- Criar uma frota para executar o servidor do jogo.
- Conectar-se ao servidor do jogo a partir do cliente de jogo de exemplo.
- Analisar as métricas da frota e da sessão de jogo.

Após essas etapas, você pode executar vários clientes de jogos simultaneamente e jogar o jogo para gerar dados de hospedagem. Em seguida, você pode explorar o console do Amazon GameLift para visualizar seus recursos de hospedagem, monitorar métricas e experimentar maneiras de escalar a capacidade de hospedagem.

Para começar, faça login no [console do Amazon GameLift](#).

Exemplo de jogo de servidores em tempo real

Este exemplo é um jogo multijogador completo chamado Mega Frog Race, com código-fonte incluído. O exemplo mostra como integrar seu cliente de jogo aos servidores em tempo real. Você também pode usar esse exemplo de jogo como ponto de partida para experimentar outros recursos do Amazon GameLift, como o FlexMatch.

Para obter um tutorial prático, consulte [Criar servidores para jogos móveis multijogador com apenas algumas linhas de JavaScript](#) no blog da AWS para jogos.

Para o código-fonte do Mega Frog Race, consulte o [repositório do GitHub](#).

O código-fonte inclui as seguintes partes:

- Cliente de jogo – Um código-fonte para o cliente de jogo C++, criado no Unity. O cliente do jogo obtém informações de conexão da sessão do jogo, se conecta ao servidor e troca atualizações com outros jogadores.
- Serviço de back-end – Um código-fonte para uma função do AWS Lambda que gerencia chamadas diretas de API para o Amazon GameLift.
- Script em tempo real – Um arquivo de script de origem que configura uma frota de servidores em tempo real para o jogo. Esse script inclui a configuração mínima necessária para que os servidores em tempo real se comuniquem com o Amazon GameLift e hospedem jogos.

Roteiro de hospedagem gerenciada do Amazon GameLift

Este tópico ajuda você a escolher entre as diferentes opções de hospedagem do Amazon GameLift para seu jogo multijogador baseado em sessão. O restante dos tópicos desta seção explica como usar o Amazon GameLift para sua hospedagem gerenciada.

Antes de começar a se preparar para lançar seu jogo em produção, preencha o questionário de lançamento para começar a trabalhar com a equipe do Amazon GameLift.

Tópicos

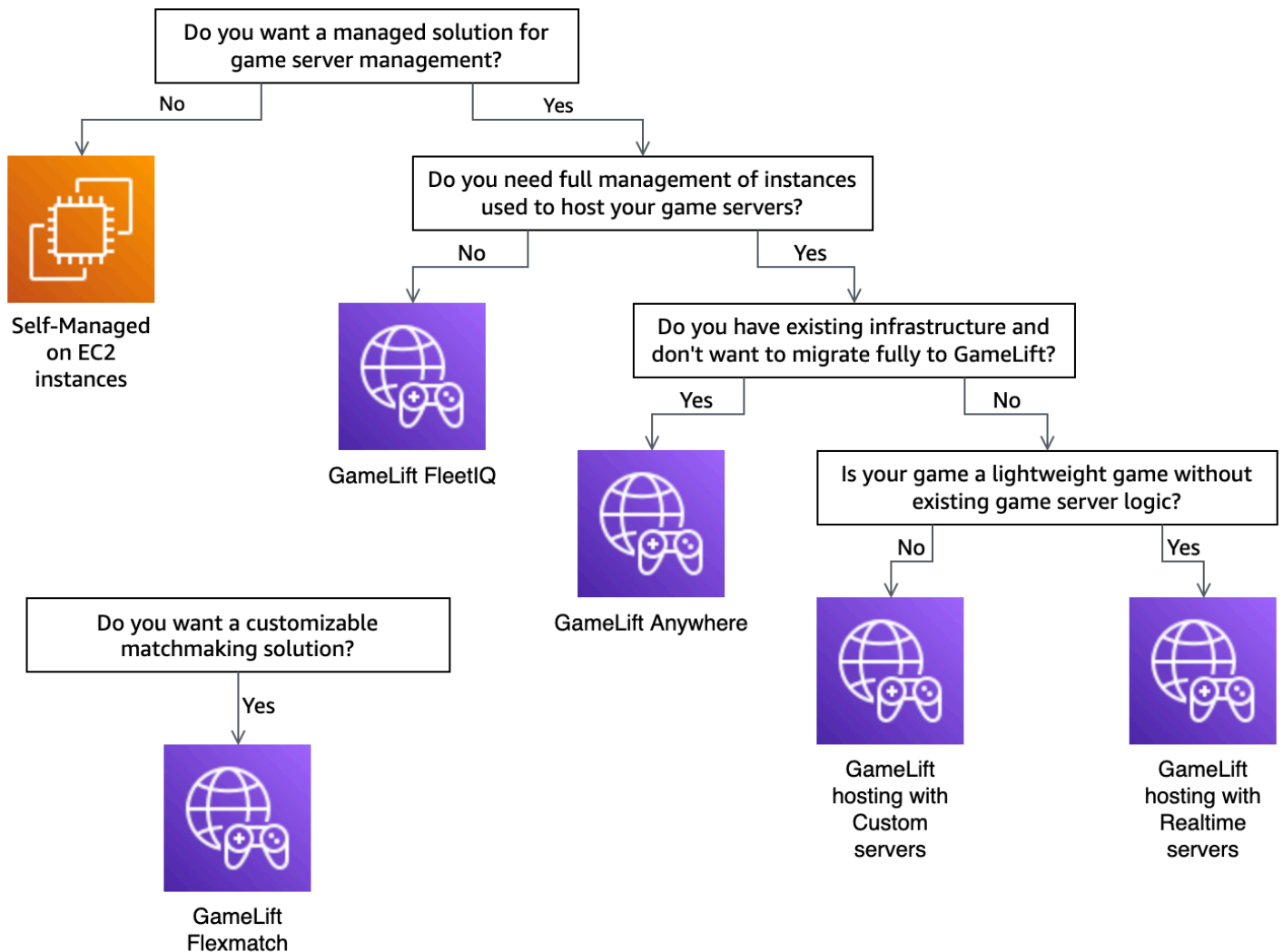
- [Escolha uma opção de hospedagem](#)
- [Preparar seu jogo para o Amazon GameLift](#)
- [Teste sua integração com o Amazon GameLift](#)
- [Planeje e implante seus recursos do Amazon GameLift](#)
- [Projete seu serviço de cliente de jogos](#)
- [Configure métricas e registros para o Amazon GameLift](#)
- [Listas de verificação de lançamento de jogos](#)

Escolha uma opção de hospedagem

O fluxograma a seguir faz perguntas para levar você à solução correta do Amazon GameLift, para seu caso de uso.

1. Você deseja uma solução gerenciada para gerenciamento de servidores de jogos?
 - Sim – Continue com a etapa dois.
 - Não – Considere servidores de jogos autogerenciados em instâncias do Amazon EC2.
2. Você precisa de controle total das instâncias que hospedam seus servidores de jogos?
 - Sim – Considere o Amazon GameLift FleetIQ.
 - Não – Continue na etapa 3.
3. Você tem uma infraestrutura existente que deseja usar com o Amazon GameLift?
 - Sim – Considere o Anywhere do Amazon GameLift.

- Não – Continue com a etapa quatro.
4. Seu jogo é leve sem a lógica existente do servidor de jogos?
- Sim – Considere os servidores em tempo real.
 - Não – Considere servidores personalizados.



Aqui estão mais algumas informações sobre algumas das opções de hospedagem do Amazon GameLift mencionadas no fluxograma:

Anywhere do Amazon GameLift

Use o Anywhere do Amazon GameLift para hospedar seus jogos em seu próprio hardware com o benefício das ferramentas de gerenciamento do Amazon GameLift. Também é possível

usar frotas Anywhere para testar seus servidores de jogos de forma iterativa. Para obter mais informações, consulte [Crie uma GameLift Anywhere frota da Amazon](#).

Amazon GameLift gerenciado

Há duas opções de hospedagem gerenciada do Amazon GameLift:

Servidores personalizados – O Amazon GameLift hospeda seu servidor personalizado que executa o binário do seu servidor de jogos.

Servidores em tempo real – O Amazon GameLift hospeda seu servidor de jogos leve.

Amazon GameLift FleetIQ

No fluxograma, uma migração mover sem alterações (lift-and-shift) se refere a uma migração quando você não pode fazer alterações na arquitetura do jogo. Usar o Amazon GameLift FleetIQ exige menos alterações em sua implantação atual e fornece ferramentas do Amazon GameLift para gerenciamento de frotas. Para obter mais informações, consulte o [Guia do desenvolvedor do Amazon GameLift FleetIQ](#).

Caso decida usar o Anywhere do Amazon GameLift ou o Amazon GameLift gerenciado, continue para [Preparar seu jogo para o Amazon GameLift](#).

Preparar seu jogo para o Amazon GameLift

Este tópico descreve as etapas para preparar seu jogo multijogador para integração com a hospedagem gerenciada do Amazon GameLift. Para preparar seu jogo, ative a comunicação entre ele e o Amazon GameLift.

Prepare o servidor de jogos personalizado

Para iniciar e interromper sessões de jogos e executar outras tarefas, um servidor de jogos deve poder notificar o Amazon GameLift sobre seu status. Para ativar a comunicação com o Amazon GameLift, adicione código ao seu projeto de servidor de jogos. Para obter mais informações, consulte [Integre jogos com servidores de jogos personalizados](#).

1. Prepare o servidor de jogos personalizado para hospedagem no Amazon GameLift..
 - Obtenha o [SDK de servidor do Amazon GameLift](#) e compile-o para sua linguagem de programação e mecanismo de jogos preferidos.

- Adicione código ao seu projeto de servidor de jogo para ativar a comunicação com o Amazon GameLift.
2. Prepare seu cliente de jogo para se conectar a sessões de jogos hospedadas pelo Amazon GameLift.
 - Adicione o SDK da AWS ao serviço de back-end e ao projeto do cliente do jogo. Para obter mais informações, consulte [Baixar os SDKs do Amazon GameLift para serviços ao cliente](#).
 - Adicione funcionalidade para recuperar informações sobre sessões de jogos, realizar novas sessões de jogos e reservar espaço para jogadores em uma sessão de jogo.
 - (Opcional) Use o FlexMatch para a organização de partidas de jogadores. Para obter mais informações, consulte [Integração do FlexMatch com a hospedagem do Amazon GameLift](#).

Preparar seu servidor em tempo real

O Servidores em Tempo Real do Amazon GameLift oferece uma solução de servidor leve que você poderá configurar para se adequar ao seu jogo. Um servidor em tempo real oferece os mesmos benefícios que o Amazon GameLift oferece aos servidores de jogos, mas com capacidade reduzida de personalização do servidor de jogos.

Crie um script em tempo real para hospedagem no Amazon GameLift.

Os scripts em tempo real contêm a configuração do servidor e a lógica de jogo personalizada opcional. Os servidores em tempo real são criados para iniciar e interromper sessões de jogo, aceitar conexões de jogadores e gerenciar a comunicação com o Amazon GameLift e entre jogadores em um jogo. Também existem ganchos para você adicionar uma lógica de servidor personalizada ao seu jogo. Os servidores em tempo real usam Node.js e JavaScript. Para obter mais informações, consulte [Criar um script em tempo real](#) e [Teste sua integração com o Amazon GameLift](#).

Teste sua integração com o Amazon GameLift

O Amazon GameLift oferece suporte à iteração rápida ao testar seus servidores de jogos. Este tópico orienta você pelos tipos de testes disponíveis.

Servidores de jogos personalizados

Use o Amazon GameLift para integrar hardware em qualquer lugar do seu ambiente à sua arquitetura de hospedagem de jogos do Amazon GameLift. O Anywhere do Amazon GameLift registra seu hardware no Amazon GameLift em uma frota Anywhere, para que você possa testar

usando seu próprio computador de desenvolvimento local. Para obter mais informações sobre testes com o Anywhere do Amazon GameLift, consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#). Para obter mais informações sobre o uso do Anywhere Amazon GameLift para hospedar seus jogos com soluções on-premises, consulte [Escolhendo os GameLift recursos computacionais da Amazon](#).

Servidores em tempo real

Com os servidores em tempo real, atualize seus scripts a qualquer momento. Ao atualiza um script em tempo real, o Amazon GameLift distribui a nova versão para seus recursos de hospedagem em minutos. Após o Amazon GameLift implantar o novo script, todas as novas sessões de jogo usam a nova versão do script. Após o Amazon GameLift implantar o novo script, você poderá começar a testar imediatamente. Para obter mais informações sobre os servidores em tempo real, consulte [Integração de jogos com Servidores em Tempo Real do Amazon GameLift](#)

Planeje e implante seus recursos do Amazon GameLift

Use as dicas a seguir para ajudar a planejar sua implantação global de recursos do Amazon GameLift. Para obter informações sobre onde você pode hospedar seus jogos com o Amazon GameLift, consulte [Locais GameLift de hospedagem da Amazon](#).

Para implantar os recursos do Amazon GameLift, execute as seguintes tarefas:

- Empacote e faça o upload do seu servidor de jogos para o Amazon GameLift ou para o seu hardware. Ao fazer o upload do seu servidor para o Amazon GameLift, você o carrega somente para a Região da AWS de origem da sua frota. O Amazon GameLift distribui automaticamente o servidor para outros locais da frota. Para obter mais informações, consulte [Fazer upload de compilações e scripts para o Amazon GameLift](#).
- Projete e implante uma frota do Amazon GameLift para seu jogo. Determine o tipo de recursos de computação a serem usados, em quais locais implantar, se deseja usar filas e outras opções. Para obter mais informações, consulte [Guia de criação de frota do Amazon GameLift](#).
- Crie filas para gerenciar novos posicionamentos de sessões de jogo e estratégias de Instância Spot. Para obter mais informações, consulte [Criar uma fila de sessões de jogo](#).
- Use o ajuste de escala automático para gerenciar a capacidade de hospedagem da sua frota para a demanda esperada dos jogadores. Para obter mais informações, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#).
- Use as regras de marcação de jogos do FlexMatch para o seu jogo. Para obter mais informações, consulte [Integração do FlexMatch com a hospedagem do Amazon GameLift](#).

Implante automaticamente seus recursos do Amazon GameLift

Para simplificar a implantação global dos recursos do Amazon GameLift, é recomendável usar a [infraestrutura como código \(IaC\)](#) para definir os recursos. Como o AWS CloudFormation do Amazon GameLift oferece suporte a modelos, é possível definir parâmetros nos modelos para qualquer configuração específica da implantação.

Para gerenciar a implantação de suas pilhas AWS CloudFormation, também é recomendável o uso de ferramentas e serviços de integração e entrega contínuas (CI/CD), como AWS CodePipeline. Eles ajudam você a implantar automaticamente ou com aprovação sempre que você cria o binário do servidor de jogos.

A seguir estão algumas etapas comuns da implantação de recursos do Amazon GameLift para uma nova versão do servidor de jogos que poderá automatizar usando uma ferramenta ou serviço de CI/CD:

- Compilar e testar o binário do seu servidor de jogos.
- Fazer o upload do binário para o Amazon GameLift ou para o seu hardware.
- Implantação de novas frotas na nova compilação.
- Depois de implantar as novas frotas, remova as frotas da versão anterior da fila do Amazon GameLift e as substitua pelas novas frotas.
- Após a versão anterior, as frotas terminam com sucesso todas as sessões de jogo, excluindo as pilhas AWS CloudFormation dessas frotas.

Também é possível usar o AWS Cloud Development Kit (AWS CDK) para definir os recursos do Amazon GameLift. Para obter mais informações sobre o AWS CDK, consulte o [Guia do desenvolvedor do AWS Cloud Development Kit \(AWS CDK\)](#).

Projete seu serviço de cliente de jogos

É recomendável implementar um serviço de cliente de jogos que autentique seus jogadores e se comunique com a API do Amazon GameLift. Ao implementar um serviço personalizado de cliente de jogos, você poderá:

- Personalizar a autenticação para os jogadores.
- Controlar como o Amazon GameLift combina e inicia as sessões de jogo.

- Usar o banco de dados de jogadores para atributos de jogadores, como classificação de habilidade para marcação de jogos, em vez de confiar no cliente.

Usar um serviço de cliente de jogos também reduz os riscos de segurança introduzidos por clientes de jogos que interagem diretamente com a API do Amazon GameLift.

Autenticar os jogadores

Use o Amazon Cognito e os IDs de sessão do jogador para autenticar os clientes do jogo. Para gerenciar o ciclo de vida e as propriedades das identidades de jogadores, use os grupos de usuários do Amazon Cognito.

Se preferir, crie uma solução de identidade personalizada e hospede-a na AWS. Também é possível usar os autorizadores Lambda para uma lógica de autorização personalizada com a API Gateway.

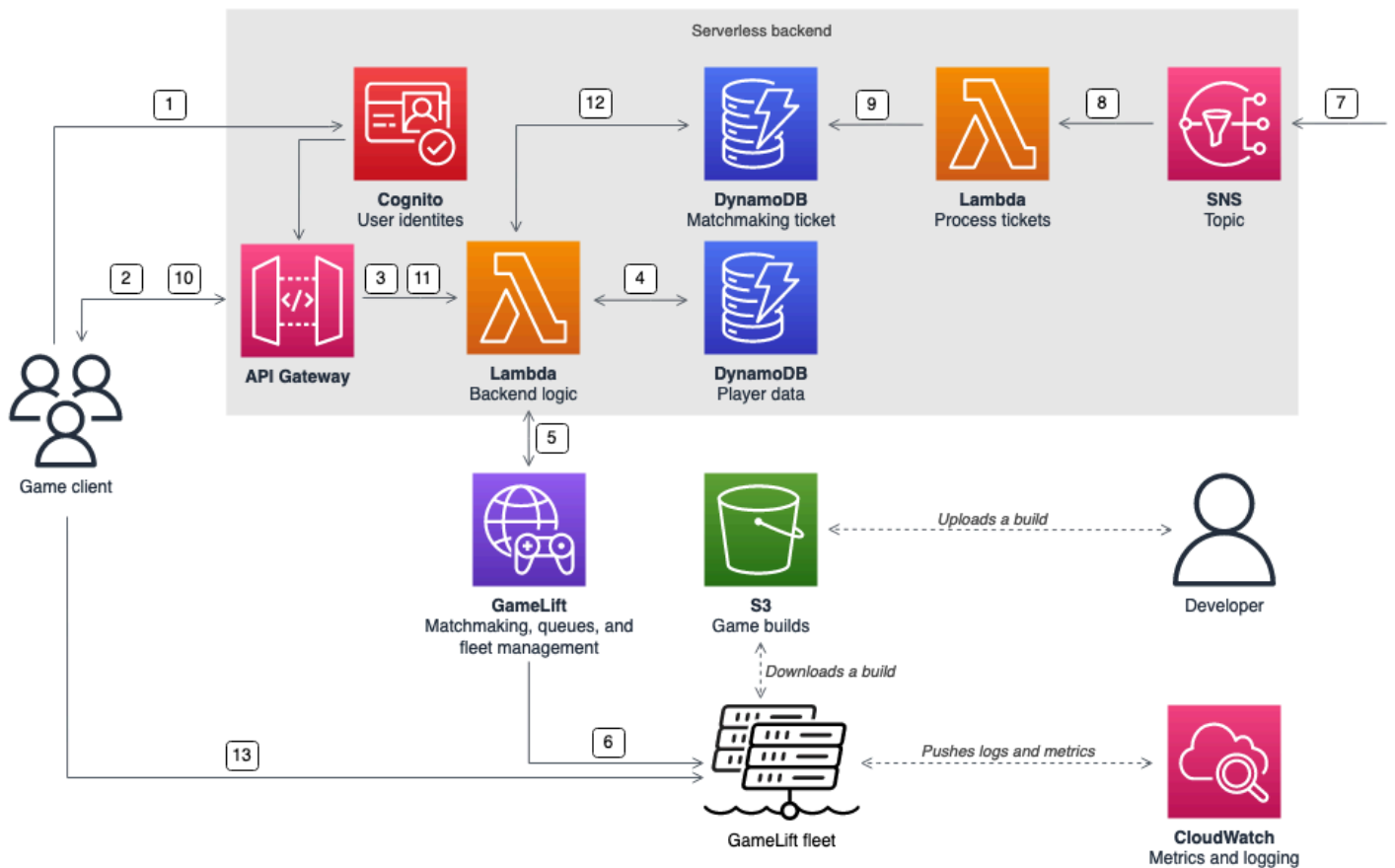
Recursos adicionais:

- [Usar banco de identidades \(identidades federadas\)](#) (Guia do desenvolvedor do Amazon Cognito)
- [Introdução aos grupos de usuários](#) (Guia do Desenvolvedor do Amazon Cognito)
- [Como configurar a autenticação do jogador com o Amazon Cognito](#) (para blogs de jogos da AWS)

Servidores de sessão de jogo autônomos com um back-end de tecnologia sem servidor

Ao usar uma arquitetura de serviço ao cliente de tecnologia sem servidor, o back-end pode visualizar o status dos tickets de marcação de jogos em um banco de dados altamente escalável, em vez de acessar diretamente a API do Amazon GameLift.

O diagrama a seguir mostra um back-end de tecnologia sem servidor criado com Serviços da AWS, para combinar jogadores com jogos executados em frotas do Amazon GameLift. A lista a seguir fornece uma descrição para cada texto explicativo numerado no diagrama. Para testar este exemplo, consulte [Hospedagem de jogos baseada em sessão multijogador no GitHub da AWS](#).



1. O cliente do jogo solicita uma identidade de usuário do Amazon Cognito a partir de um banco de identidades do Amazon Cognito.
2. O cliente do jogo recebe credenciais de acesso temporárias e solicita uma sessão de jogo por meio de uma API do Amazon API Gateway.
3. O API Gateway invoca uma função AWS Lambda.
4. A função do Lambda solicita dados do player de uma tabela NoSQL do Amazon DynamoDB. A função fornece a identidade do Amazon Cognito nos dados de contexto da solicitação.
5. A função do Lambda solicita uma correspondência por meio de marcação de jogos do Amazon GameLift FlexMatch.
6. O FlexMatch combina um grupo de jogadores com latência adequada e, em seguida, solicita a colocação de uma sessão de jogo por meio de uma fila do Amazon GameLift. A fila tem frotas com um ou mais Região da AWS locais nela.
7. Depois que o Amazon GameLift coloca a sessão em um dos locais da frota, o Amazon GameLift envia uma notificação de evento a um tópico do Amazon Simple Notification Service (Amazon SNS).

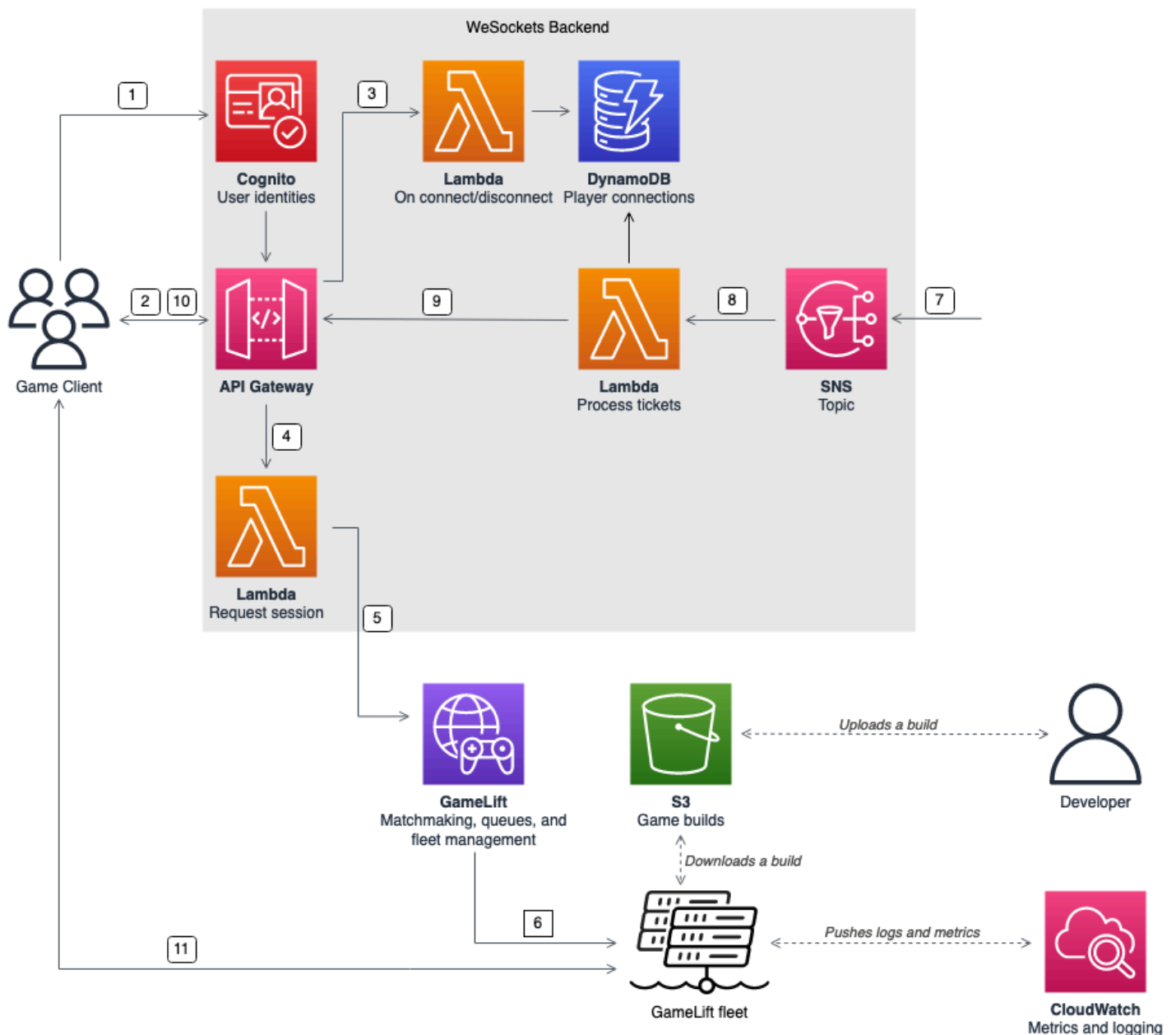
8. Uma função do Lambda recebe o evento do Amazon SNS e o processa.
9. Se o ticket de marcação de jogos for um evento de MatchmakingSucceeded, a função do Lambda grava o resultado, junto com a porta e o endereço IP do servidor do jogo, em uma tabela do DynamoDB.
10. O cliente do jogo faz uma solicitação assinada ao API Gateway para visualizar o status do ticket de marcação de jogos em um intervalo específico.
11. O API Gateway usa uma função do Lambda que verifica o status do ticket de marcação de jogos.
12. A função do Lambda verifica a tabela do DynamoDB para visualizar se o ticket foi bem-sucedido. Se for bem-sucedida, a função envia a porta e o endereço IP do servidor do jogo, junto com o ID da sessão do jogador, de volta ao cliente. Se o ticket não for bem-sucedido, a função enviará uma resposta verificando se a partida ainda não está pronta.
13. O cliente do jogo se conecta ao servidor do jogo usando TCP ou UDP usando a porta e o endereço IP fornecidos pelo serviço de back-end. O cliente do jogo então envia o ID da sessão do jogador para o servidor do jogo, que então valida o ID usando o Server SDK do Amazon GameLift.

Servidores de sessão de jogo autônomos com um back-end baseado em WebSocket

Ao usar uma arquitetura baseada em WebSockets do Amazon API Gateway, é possível fazer solicitações de marcação de jogos com WebSockets e enviar notificações push para conclusão de marcação de jogos usando mensagens iniciadas pelo servidor. Essa arquitetura melhora o desempenho por meio da comunicação bidirecional entre o cliente e o servidor.

Para obter mais informações sobre o uso das APIs WebSock do API Gateway, consulte [Trabalhar com APIs WebSocket](#).

O diagrama a seguir mostra uma arquitetura de back-end baseada em WebSocket que usa o API Gateway e outros Serviços da AWS para combinar jogadores em jogos executados nas frotas do Amazon GameLift. A lista a seguir fornece uma descrição para cada texto explicativo numerado no diagrama.



1. O cliente do jogo solicita uma identidade de usuário do Amazon Cognito a partir de um banco de identidades do Amazon Cognito.
2. O cliente do jogo assina uma conexão de WebSocket com uma API do API Gateway com as credenciais do Amazon Cognito.
3. O API Gateway chama uma função AWS Lambda na conexão. A função armazena as informações de conexão em uma tabela do Amazon DynamoDB.
4. O cliente do jogo envia uma mensagem para uma função do Lambda, por meio da API do API Gateway pela conexão de WebSocket, para solicitar uma sessão.

5. Uma função do Lambda recebe a mensagem e, em seguida, solicita uma correspondência por meio de marcação de jogos do Amazon GameLift FlexMatch.
6. Depois que o FlexMatch combina um grupo de jogadores, o FlexMatch solicita a colocação de uma sessão de jogo por meio de uma fila do Amazon GameLift.
7. Após o Amazon GameLift colocar a sessão em um dos locais da frota, o Amazon GameLift envia uma notificação de evento a um tópico do Amazon Simple Notification Service (Amazon SNS).
8. Uma função do Lambda recebe o evento do Amazon SNS e o processa.
9. Se o ticket de marcação de jogos for um evento de MatchmakingSucceeded, a função do Lambda solicitará a conexão correta do jogador do DynamoDB. Em seguida, a função envia uma mensagem para o cliente do jogo por meio da API do API Gateway pela conexão de WebSocket. Nessa arquitetura, o cliente do jogo não pesquisa ativamente o status de marcação de jogos.
10. O cliente do jogo recebe a porta e o endereço IP do servidor do jogo, junto com o ID da sessão do jogador, por meio da conexão de WebSocket.
11. O cliente do jogo se conecta ao servidor do jogo usando TCP ou UDP usando a porta e o endereço IP fornecidos pelo serviço de back-end. O cliente do jogo também envia o ID da sessão do jogador para o servidor do jogo, que então valida o ID usando o Server SDK do Amazon GameLift.

Configure métricas e registros para o Amazon GameLift

Use os dados coletados dos servidores e recursos de jogos do Amazon GameLift para ajudar a identificar anomalias. Também é possível usar métricas para ajudar a melhorar o desempenho.

As principais áreas a serem observadas no Amazon GameLift incluem:

- Métricas do serviço Amazon GameLift – O Amazon GameLift fornece métricas do Amazon CloudWatch sobre seus recursos, incluindo servidores de jogos, frotas, filas e FlexMatch. Encontre essas métricas no console do Amazon GameLift e no console do CloudWatch. Para obter mais informações sobre métricas do Amazon GameLift no CloudWatch, consulte [Monitorar o Amazon GameLift com o Amazon CloudWatch](#).
- Métricas do servidor de jogos – O Amazon GameLift não pode acessar as métricas do seu servidor de jogos. No entanto, você poderá enviar métricas personalizadas para o CloudWatch diretamente do seu servidor de jogos usando o agente do CloudWatch. Também é possível usar o perfil do (IAM) da frota AWS Identity and Access Management e o SDK da AWS para enviar métricas diretamente para o CloudWatch. Para visualizar um exemplo de como configurar métricas,

consulte [Hospedagem de jogos baseada em sessões multijogador no AWS](#) no GitHub. Esse repositório inclui um exemplo de configuração e código do atendente do CloudWatch para um cliente C# StatsD.

- Registros do servidor do jogo – Para configurar seus arquivos de log do servidor de jogos no servidor do jogo, use a configuração do Server SDK do Amazon GameLift. Também é possível usar o Amazon CloudWatch Logs como uma solução de gerenciamento de logs em tempo real e pode configurar logs com o atendente do CloudWatch. Para obter mais informações, consulte [Registro de mensagens do servidor no Amazon GameLift](#).

Listas de verificação de lançamento de jogos

Use essas listas de verificação para validar as fases de implantação do jogo. Nas listas de verificação, os itens marcados como [Críticos] são essenciais para o lançamento da produção.

Tópicos

- [Integração](#)
- [Testar](#)
- [Executar](#)
- [Pós-lançamento](#)

Integração

Use a lista de verificação a seguir para rastrear itens para integrar o jogo à hospedagem do Amazon GameLift. Os itens marcados como [Críticos] são essenciais para o lançamento da produção.

- [Crítico] Preencha o questionário de integração do Amazon GameLift no [console do Amazon GameLift](#).
- [Crítico] [Projete e implemente um serviço de back-end](#) para clientes de jogos interagirem com seus servidores de jogos.
- [Crítico] [Crie funções de AWS Identity and Access Management \(IAM\)](#) que você fornece às instâncias do servidor Amazon GameLift para acesso a outros recursos da AWS.
- [Crítico] [Projete e implemente o failover para outras Regiões da AWS](#) para o FlexMatch e filas.
- [Planeje a distribuição de frotas para os locais de destino](#), considerando a fila e a estrutura da frota do seu jogo.

- [Automatize sua implantação](#) usando infraestrutura como código (IaC) com o AWS CloudFormation e o AWS Cloud Development Kit (AWS CDK).
- [Colete registros e análises](#) usando o Amazon CloudWatch e o Amazon Simple Storage Service (Amazon S3).

Testar

Use a lista de verificação a seguir para rastrear itens de teste ao desenvolver o jogo com a hospedagem do Amazon GameLift. Os itens marcados como [Críticos] são essenciais para o lançamento da produção.

- [Crítico] Preencha o questionário de lançamento e envie o questionário preenchido para a equipe de lançamento do Amazon GameLift. Encontre o questionário de lançamento no [console do Amazon GameLift](#).
- [Crítico] [Solicite aumentos de cotas de serviços do Amazon GameLift](#) e outras cotas do AWS service (Serviço da AWS) para que seu ambiente ativo possa ser escalado de acordo com as necessidades de produção.
- [Crítico] Verifique se as portas abertas nas frotas ativas correspondem ao intervalo de portas que os servidores poderiam usar.
- [Crítico] Feche a porta RDP 3389 e a porta SSH 22.
- Desenvolva um plano para o gerenciamento de DevOps do jogo. Caso esteja usando o Amazon CloudWatch Logs ou métricas personalizadas do Amazon CloudWatch, defina alarmes para problemas graves ou críticos na frota de servidores. Simule falhas e teste os runbooks.
- [Verifique se o número de servidores](#) em execução em uma instância em uso total está dentro dos recursos do tipo de instância do servidor.
- [Ajuste sua política de escalabilidade](#) para ser mais conservadora no início e fornecer mais capacidade ociosa do que considera ser necessário. É possível otimizar o custo posteriormente. Considere o uso de uma política de escalabilidade com base em destino com 20% de capacidade ociosa.
- [Use as regras de latência do FlexMatch](#) para combinar jogadores que estão geograficamente próximos das mesmas Região da AWS. Teste como isso se comporta sob carga com dados de latência sintéticos do seu cliente de teste de carga.
- Teste a carga da autenticação do jogador e da infraestrutura da sessão de jogo para observar se ela é dimensionada de forma eficaz para atender à demanda.
- Verifique se um servidor deixado em execução por vários dias ainda pode aceitar conexões.

- Eleve o nível do seu plano AWS Support para Business ou Enterprise para que a AWS possa responder a você durante problemas ou interrupções.

Executar

Use a lista de verificação a seguir para rastrear os itens de lançamento do seu jogo hospedado no Amazon GameLift. Os itens marcados como [Críticos] são essenciais para o lançamento da produção.

- [Crítico] [Defina a política de proteção da frota](#) para proteção total em todas as frotas ativas, para que a redução não interrompa as sessões ativas do jogo.
- [Crítico] [Defina tamanhos máximos de frota](#) altos o suficiente para acomodar o pico de demanda previsto, no mínimo. É recomendável dobrar seu tamanho máximo para uma demanda imprevista.
- Incentive toda a sua equipe de desenvolvimento a participar do evento de lançamento e monitorar o lançamento do jogo em uma sala de lançamento.
- Monitore a latência e a experiência do jogador.

Pós-lançamento

Use a lista de verificação a seguir para rastrear itens pós-lançamento do seu jogo hospedado no Amazon GameLift.

- [Ajuste as regras de escalabilidade para minimizar a capacidade ociosa.](#)
- [Modifique as regras do FlexMatch](#) ou [adicione outros locais](#) com base em seus requisitos de latência.
- Otimize o executável do servidor, pois sua eficiência de desempenho afeta diretamente os custos da frota. Para executar mais sessões de jogo com a mesma infraestrutura, aumente o número de processos do servidor por instância.
- [Use seus dados de análise](#) para impulsionar o desenvolvimento contínuo, melhorar a experiência do jogador e a longevidade do jogo e otimizar a monetização.

Preparando jogos para a Amazon GameLift

Para preparar seu jogo multijogador para hospedagem na Amazon GameLift, configure a comunicação entre seu jogo e a Amazon GameLift. Os tópicos desta seção fornecem ajuda detalhada para integrar seu jogo com a Amazon GameLift, servidores de jogos personalizados e servidores em tempo real, e para adicionar matchmaking com FlexMatch

Tópicos

- [Integre jogos com servidores de jogos personalizados](#)
- [Integração de jogos com Servidores em Tempo Real do Amazon GameLift](#)
- [Integrando jogos com o GameLift plugin Amazon para Unity](#)
- [Integrando jogos com o GameLift plugin Amazon para Unreal Engine](#)
- [Obtenha dados da frota para uma instância do Amazon GameLift](#)
- [Adicionar a marcação de jogos do FlexMatch](#)

Integre jogos com servidores de jogos personalizados

O Amazon GameLift fornece um conjunto de ferramentas completo para preparar jogos multijogador e servidores de jogos personalizados para execução no Amazon GameLift. Os SDKs do Amazon GameLift contêm as bibliotecas necessárias para que clientes e servidores de jogos se comuniquem com o Amazon GameLift. Para obter mais informações sobre os SDKs e onde obtê-los, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).

Os tópicos desta seção contêm instruções detalhadas sobre como adicionar a funcionalidade do Amazon GameLift ao cliente e ao servidor do jogo antes de implantar no Amazon GameLift. Para obter um roteiro completo para a execução de seu jogo no Amazon GameLift, consulte [Roteiro de hospedagem gerenciada do Amazon GameLift](#).

Tópicos

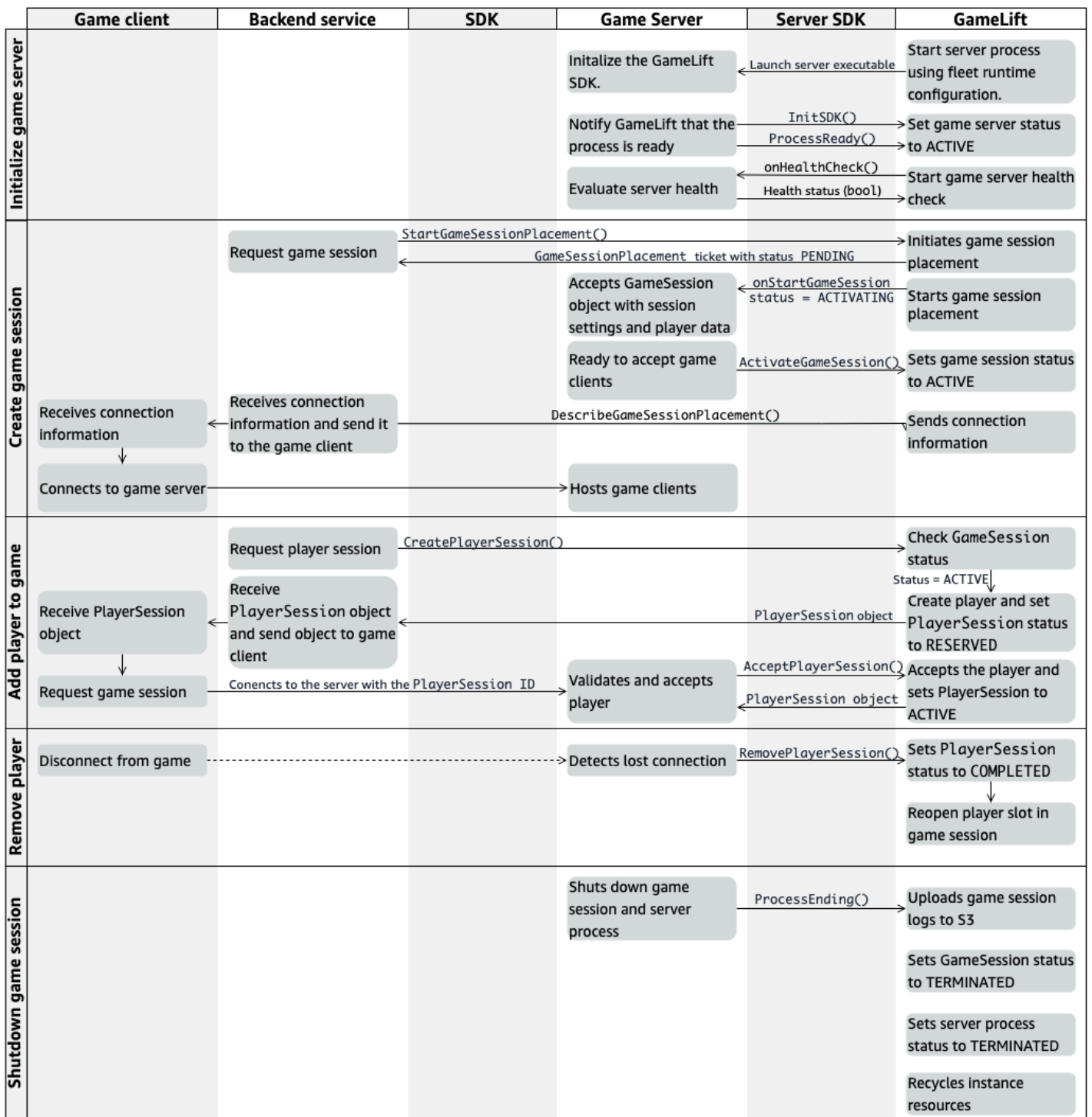
- [Amazon GameLift e interações entre cliente e servidor de jogos](#)
- [Integrar os servidor de jogos ao Amazon GameLift](#)
- [Integre o cliente de jogo com o Amazon GameLift](#)
- [Mecanismos de jogo e Amazon GameLift](#)
- [Teste sua integração usando frotas Anywhere do Amazon GameLift](#)

- [Teste sua integração usando o Amazon GameLift Local](#)

Amazon GameLift e interações entre cliente e servidor de jogos

Este tópico descreve as interações entre o cliente do jogo, um serviço de back-end, um servidor de jogos e o Amazon GameLift.

O diagrama a seguir ilustra as interações entre o cliente do jogo, o serviço de back-end, o SDK do Amazon GameLift, o servidor de jogos EC2 gerenciado, o SDK do servidor Amazon GameLift e o Amazon GameLift. Para obter uma descrição detalhada das interações mostradas, consulte as seções a seguir nesta página.



Inicializar um servidor de jogos

As etapas a seguir descrevem as interações que ocorrem ao preparar o servidor de jogos para hospedar sessões de jogo.

1. O Amazon GameLift inicia o servidor executável em uma instância do Amazon Elastic Compute Cloud (Amazon EC2).
2. O servidor do jogo chama:
 - a. `InitSDK()` para inicializar o SDK do servidor.
 - b. `ProcessReady()` para comunicar a preparação da sessão de jogo, informações de conexão e localização dos arquivos de log da sessão de jogo.

O processo do servidor então espera por um retorno de chamada do Amazon GameLift.

3. O Amazon GameLift atualiza o status do processo do servidor como ACTIVE, para permitir a colocação da sessão de jogo.
4. O Amazon GameLift começa a chamar o retorno de chamada `onHealthCheck` e continua a chamá-lo periodicamente enquanto o processo do servidor está ativo. O processo do servidor pode informar íntegro ou não dentro de um minuto.

Criar uma sessão de jogo

Após inicializar o servidor de jogos, as seguintes interações ocorrem ao criar sessões de jogo para hospedar os jogadores.

1. O serviço de back-end chama a operação `StartGameSessionPlacement()` do SDK.
2. O Amazon GameLift cria um novo ticket `GameSessionPlacement` com status PENDING e o devolve ao serviço de back-end.
3. O serviço de back-end obtém o status do tíquete de colocação em uma fila. Para obter mais informações, consulte [Configurar notificações de eventos para o posicionamento de sessões de jogo](#).
4. O Amazon GameLift inicia o posicionamento da sessão de jogo selecionando uma frota apropriada e procurando por um processo de servidor ativo em uma frota com sessões de jogo 0. Quando o Amazon GameLift localiza um processo de servidor, o Amazon GameLift faz o seguinte:
 - a. Cria um objeto de `GameSession` com as configurações da sessão de jogo e os dados do jogador da solicitação de posicionamento com um status ACTIVATING.

- b. Invoca o retorno de chamada `onStartGameSession` no processo do servidor. O Amazon GameLift passa informações para o objeto de `GameSession` indicando que o processo do servidor pode configurar a sessão do jogo.
 - c. Altera o número de sessões de jogo do processo do servidor para 1.
5. O processo do servidor executa a função de retorno de chamada `onStartGameSession`. Quando o processo do servidor estiver pronto para aceitar conexões de jogadores, ele chama a `ActivateGameSession()` e aguarda por conexões de jogadores.
6. O Amazon GameLift atualiza o objeto de `GameSession` com informações de conexão para o processo do servidor. (Essas informações incluem a configuração de porta com a qual foi relatada com `ProcessReady()`.) O Amazon GameLift também altera o status para `ACTIVE`.
7. O serviço de back-end chama o `DescribeGameSessionPlacement()` para detectar o status atualizado do ticket. O serviço de back-end então usa as informações de conexão para conectar o cliente do jogo ao processo do servidor e ingressar na sessão do jogo.

Adicionar um jogador a um jogo

Essa sequência descreve o processo de adicionar um jogador a uma sessão de jogo existente. As sessões de jogadores também podem ser solicitadas como parte de uma solicitação de colocação de sessão de jogo.

1. O serviço de back-end chama a operação `CreatePlayerSession()` da API do cliente com um ID de sessão do jogo.
2. O Amazon GameLift verifica o status da sessão do jogo (deve ser `ACTIVE`) e procura um slot de jogador aberto na sessão de jogo. Se estiver disponível, o Amazon GameLift fará o seguinte:
 - a. Cria um objeto `PlayerSession` e define o status como `RESERVED`.
 - b. Responde à solicitação de serviço de back-end com o objeto de `PlayerSession`.
3. O serviço de back-end conecta o cliente do jogo diretamente ao processo do servidor com o ID da sessão do jogador.
4. O servidor chama a operação de `AcceptPlayerSession()` da API do servidor para validar o ID da sessão do jogador. Se validado, o Amazon GameLift transmitirá o objeto de `PlayerSession` ao processo do servidor. O processo do servidor aceita ou rejeita a conexão.
5. O Amazon GameLift executará um dos seguintes procedimentos:

- a. Se a conexão for aceita, o Amazon GameLift definirá o status `PlayerSession` como `ACTIVE`.
- b. Se nenhuma resposta for recebida dentro de 60 segundos após a chamada `CreatePlayerSession()` original do servidor back-end, o Amazon GameLift alterará o status `PlayerSession` para `TIMEDOUT` e reabrirá o slot do jogador na sessão do jogo.

Remover um jogador

Ao remover jogadores de uma sessão de jogo para criar espaço para novos jogadores entrarem, as seguintes interações ocorrem.

1. Um jogador se desconecta do jogo.
2. O servidor detecta a conexão perdida e chama a operação de `RemovePlayerSession()` da API do servidor.
3. O Amazon GameLift altera o status do `PlayerSession` para `COMPLETED` e reabre o espaço do jogador na sessão de jogo.

Encerre a sessão de jogo

Essa sequência de interações ocorre quando um processo do servidor encerra a sessão atual do jogo.

1. O servidor encerra a sessão do jogo e o servidor.
2. O servidor chama `ProcessEnding()` para o Amazon GameLift.
3. O Amazon GameLift faz o seguinte:
 - a. Carrega os logs de sessão de jogo no Amazon Simple Storage Service (Amazon S3).
 - b. Altera o status `GameSession` para `TERMINATED`.
 - c. Altera o status do processo do servidor para `TERMINATED`.
 - d. Recicla os recursos da instância.

Integrar os servidor de jogos ao Amazon GameLift

Depois que o servidor de jogo personalizado for implantado e executado nas instâncias do Amazon GameLift, ele deverá poder interagir com o Amazon GameLift (e potencialmente com outros recursos). Esta seção descreve como integrar o software do servidor de jogos ao Amazon GameLift.

Note

Essas instruções pressupõem que você criou uma Conta da AWS e que tem um projeto de servidor de jogo existente.

Os tópicos nesta seção descrevem como lidar com as seguintes tarefas de integração:

- Estabeleça comunicação entre o Amazon GameLift e seus servidores de jogos.
- Gere e use um certificado TLS para estabelecer uma conexão segura entre o cliente do jogo e o servidor do jogo.
- Forneça permissões para que seu software de servidor de jogos interaja com outros recursos da AWS.
- Permita que os processos do servidor de jogos obtenham informações sobre a frota em que estão executando.

Tópicos

- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Comunique-se com outros recursos AWS de suas frotas](#)

Adicione o Amazon GameLift ao servidor de jogos

Seu servidor de jogo personalizado deve se comunicar com o Amazon GameLift, porque cada processo do servidor de jogos deve poder responder aos eventos iniciados pelo Amazon GameLift. Seu servidor de jogos também deve manter o Amazon GameLift informado sobre o status do processo do servidor e as conexões dos jogadores. Para obter mais informações sobre como seu servidor de jogos, serviço de back-end, cliente de jogos e Amazon GameLift trabalham juntos para gerenciar a hospedagem de jogos, consulte [Amazon GameLift e interações entre cliente e servidor de jogos](#).

Para preparar seu servidor de jogos para interagir com o Amazon GameLift, adicione o Server SDK do Amazon GameLift ao seu projeto de servidor de jogos e integre a funcionalidade descrita neste tópico. O Server SDK está disponível em várias linguagens. Para obter mais informações sobre o Server SDK do Amazon GameLift, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).

Referências da API do Server SDK:

- [Referência do SDK 5.x do servidor Amazon GameLift para C++](#)
- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)
- [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)

Inicializar o processo do servidor

Adicione código para estabelecer comunicação com o Amazon GameLift e informar que o processo do servidor está pronto para hospedar uma sessão de jogo. Esse código deve ser executado antes de qualquer código do Amazon GameLift.

1. Inicialize o cliente da API Amazon GameLift chamando `InitSdk()`. Para inicializar um processo de servidor em um recurso de computação Anywhere do Amazon GameLift, chame `InitSdk()` com o seguinte `ServerParameters`:
 - O URL do Websocket usado para se conectar ao servidor do jogo.
 - O ID do processo usado para hospedar o servidor de jogos.
 - O ID do computador que hospeda os processos do servidor de jogos.
 - O ID da frota do GameLift contendo a computação Anywhere do Amazon GameLift.
 - O token de autorização gerado pela operação [GetComputeAuthToken](#) do Amazon GameLift.

Note

Para inicializar um servidor de jogos em uma instância do Amazon EC2 gerenciada pelo Amazon GameLift, crie os `ServerParameters` usando o construtor padrão `InitSDK()` ([C++](#)) ([C#](#)) ([Unreal](#)) (sem parâmetros). O Amazon GameLift configura o ambiente computacional e se conecta automaticamente ao Amazon GameLift para você.

2. Notifique o Amazon GameLift de que um processo do servidor está pronto para hospedar uma sessão do jogo. Chame `ProcessReady()` ([C++](#)) ([C#](#)) ([Unreal](#)) com as seguintes informações.

(Observe que você deve chamar `ProcessReady()` somente uma vez por processo do servidor).

- O número de porta usado pelo processo do servidor. O serviço de back-end fornece o número da porta e um endereço IP para que os clientes do jogo se conectem ao processo do servidor e participem de uma sessão de jogo.
- A localização dos arquivos, como registros de sessões de jogos, que você deseja que o Amazon GameLift retenha. O processo do servidor gera esses arquivos durante uma sessão de jogo. Eles são armazenados temporariamente na instância em que o processo do servidor está sendo executado e são perdidos quando a instância é encerrada. Todos os arquivos listados são carregados no Amazon GameLift. Acesse esses arquivos por meio do [console do Amazon GameLift](#) ou chamando a operação [GetGameSessionLogUrl\(\)](#) da API do Amazon GameLift.
- Os nomes das funções de retorno de chamada que o Amazon GameLift pode chamar para o processo do servidor. Seu servidor de jogo deve implementar essas funções. Para obter mais informações, consulte [\(C++\)](#) [\(C#\)](#) [\(Unreal\)](#)
 - (Opcional) `onHealthCheck` – O Amazon GameLift chama essa função regularmente para solicitar um relatório de status de integridade do servidor.
 - `onStartGameSession` – O Amazon GameLift chama essa função em resposta à solicitação do cliente [CreateGameSession\(\)](#).
 - `onProcessTerminate` – O Amazon GameLift força o processo do servidor a parar, permitindo que ele seja desligado normalmente.
 - (Opcional) `onUpdateGameSession` – O Amazon GameLift fornece um objeto de sessão do jogo atualizado para o servidor de jogos ou fornece uma atualização de status em uma solicitação de alocação de partida. O recurso de [preenchimento do FlexMatch](#) requer esse retorno de chamada.

Também é possível configurar um servidor de jogos para acessar com segurança os recursos da AWS que você possui ou controla. Para obter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

(Opcional) Reportar a integridade do processo do servidor

Adicione código ao seu servidor de jogo para implementar a função de retorno de chamada `onHealthCheck()`. O Amazon GameLift invoca esse método de retorno de chamada

periodicamente para coletar métricas de integridade. Para implementar essa função de retorno de chamada, faça o seguinte:

- Avalie o status da integridade do processo do servidor. Por exemplo, convém informar o processo do servidor como não íntegro caso alguma dependência externa falhar.
- Conclua a avaliação de integridade e responda ao retorno de chamada dentro de 60 segundos. Se o Amazon GameLift não receber uma resposta nesse período, ele considerará automaticamente o processo do servidor como não íntegro.
- Retornar um valor Booleano: verdadeiro para íntegro, falso para não íntegro.

Caso não implemente um retorno de chamada de verificação de integridade, o Amazon GameLift considerará o processo do servidor íntegro, a menos que o servidor não responda.

O Amazon GameLift usa a integridade do processo do servidor para encerrar processos não íntegros e liberar recursos. Se um processo do servidor continuar sendo não íntegro ou não responder por três verificações de integridade consecutivas, o Amazon GameLift poderá encerrar o processo e iniciar um novo. O Amazon GameLift coleta métricas sobre a integridade do processo do servidor de uma frota.

(Opcional) Obtenha um certificado TLS

e o processo do servidor estiver em execução em uma frota com a geração de certificado TLS ativada, você poderá recuperar o certificado para estabelecer uma conexão segura com um cliente de jogo e criptografar a comunicação entre cliente e servidor. Uma cópia do certificado é armazenada na instância. Para obter a localização do arquivo, chame `GetComputeCertificate()` ([C++](#)) ([C#](#)) ([Unreal](#)).

Iniciar uma sessão de jogo

Adicionar código para implementar a função de retorno de chamada `onStartGameSession`. O Amazon GameLift invoca esse retorno de chamada para iniciar uma sessão de jogo no servidor.

A função da `onStartGameSession` usa um objeto [GameSession](#) como parâmetro de entrada. Esse objeto inclui informações importantes sobre a sessão do jogo, como o número máximo de jogadores. Também pode incluir dados do jogo e dados do jogador. A implementação da função deve realizar as seguintes tarefas:

- Encerrar ações para criar uma sessão de jogo com base nas propriedades `GameSession`. No mínimo, o servidor do jogo deve associar o ID da sessão do jogo, que os clientes do jogo referenciam ao se conectar ao processo do servidor.
- Processe os dados do jogo e os dados do jogador conforme necessário. Esses dados estão no objeto de `GameSession`.
- Notifique o Amazon GameLift quando uma nova sessão de jogo estiver pronta para aceitar jogadores. Chame a operação da API do servidor `ActivateGameSession()` ([C++](#)) ([C#](#)) ([Unreal](#)). Em resposta a uma chamada bem-sucedida, o Amazon GameLift altera o status da sessão do jogo para `ACTIVE`.

(Opcional) Validar um jogador

Caso esteja monitorando o status das sessões dos jogadores, adicione um código para validar um novo jogador quando ele se conectar a um servidor de jogo. O Amazon GameLift monitora os jogadores atuais e os slots de sessão de jogo disponíveis.

Para validação, um cliente do jogo solicitando acesso à sessão do jogo deve incluir um ID da sessão do jogador. O Amazon GameLift gera automaticamente esse ID quando um jogador pede para entrar em um jogo usando [StartGameSessionPlacement\(\)](#) ou [StartMatchmaking\(\)](#). A sessão do jogador então reserva uma vaga em uma sessão de jogo.

Quando o processo do servidor do jogo recebe uma solicitação de conexão do cliente do jogo, ele chama `AcceptPlayerSession()` ([C++](#)) ([C#](#)) ([Unreal](#)) (`C++`) com o ID da sessão do jogador. Em resposta, o Amazon GameLift verifica se o ID da sessão do jogador corresponde a um slot aberto reservado na sessão do jogo. Depois que o Amazon GameLift valida o ID da sessão do jogador, o processo do servidor aceita a conexão. O jogador pode então entrar na sessão do jogo. Se o Amazon GameLift não validar o ID da sessão do jogador, o processo do servidor negará a conexão.

(Opcional) Informar o término de uma sessão de jogador

Caso esteja acompanhando o status das sessões dos jogadores, adicione um código para notificar o Amazon GameLift quando um jogador sair da sessão do jogo. Esse código deverá ser executado sempre que o processo do servidor detectar uma conexão desfeita. O Amazon GameLift usa essa notificação para rastrear os jogadores atuais e os slots disponíveis na sessão do jogo.

Para lidar com conexões perdidas, no código, adicione uma chamada à operação de API do servidor `RemovePlayerSession()` ([C++](#)) ([C#](#)) ([Unreal](#)) (`C++`) com o ID de sessão do jogador correspondente.

Encerrar uma sessão de jogo

Adicione o código à sequência de desligamento do processo do servidor para notificar o Amazon GameLift quando uma sessão de jogo estiver terminando. Para reciclar e atualizar recursos de hospedagem, o Amazon GameLift encerra os processos do servidor após a conclusão da sessão do jogo.

No início do código de desligamento do processo do servidor, chame a operação da API do servidor `ProcessEnding()` ([C++](#)) ([C#](#)) ([Unreal](#)). Essa chamada notifica o Amazon GameLift sobre o encerramento do processo do servidor. O Amazon GameLift altera o status da sessão do jogo e o status do processo do servidor para `TERMINATED`. Após chamar `ProcessEnding()`, é seguro que o processo seja encerrado.

Responder a uma notificação de desligamento do processo do servidor

Adicione o código para encerrar o processo do servidor em resposta a uma notificação do Amazon GameLift. O Amazon GameLift envia essa notificação quando o processo do servidor relata consistentemente problemas de integridade ou se a instância em que o processo do servidor está sendo executado está sendo encerrada. O Amazon GameLift pode interromper uma instância como parte de um evento de redução de capacidade ou em resposta à interrupção da Instância Spot.

Para lidar com uma notificação de desligamento, faça as seguintes alterações no código do servidor do jogo:

- Implemente a função de retorno de chamada `onProcessTerminate()`. Esta função deve chamar o código que desliga o processo do servidor. Quando o Amazon GameLift invoca essa operação, as interrupções da Instância Spot fornecem um aviso de dois minutos. Este aviso dá ao processo do servidor tempo para desconectar os jogadores, preservar dados de estado do jogo e realizar outras tarefas de limpeza.
- Chame a operação da API do servidor `GetTerminationTime()` ([C++](#)) ([C#](#)) ([Unreal](#)) a partir do código de desligamento do servidor do jogo. Se o Amazon GameLift emitiu uma chamada para interromper o processo do servidor, o `GetTerminationTime()` retornará o tempo estimado de rescisão.
- No início do código de desligamento do servidor do jogo, chame a operação da API do servidor `ProcessEnding()` ([C++](#)) ([C#](#)) ([Unreal](#)). Essa chamada notifica o Amazon GameLift de que o processo do servidor está sendo encerrado e, em seguida, o Amazon GameLift altera o status do processo do servidor para `TERMINATED`. Após chamar `ProcessEnding()`, é seguro que o processo seja encerrado.

Comunique-se com outros recursos AWS de suas frotas

Ao criar uma versão de servidor de jogos para implantação em frotas do Amazon GameLift, talvez você queira que os aplicativos em sua versão de jogo se comuniquem de forma direta e segura com outros recursos AWS que você possui. Como o Amazon GameLift gerencia suas frotas de hospedagem de jogos, você deve dar ao Amazon GameLift acesso limitado a esses recursos e serviços.

Alguns exemplos de cenários incluem:

- Use um atendente do Amazon CloudWatch para coletar métricas, logs e rastreamentos de frotas Anywhere e frotas EC2 gerenciadas
- Enviar dados de log da instância para Amazon CloudWatch Logs.
- Obtenha arquivos de jogos armazenados em um bucket do Amazon Simple Storage Service (Amazon S3).
- Ler e gravar dados de jogos (como modos de jogo ou inventário) armazenados em um banco de dados do Amazon DynamoDB ou em outro serviço de armazenamento de dados.
- Envie sinais diretamente para uma instância usando o Amazon Simple Queue Service (Amazon SQS).
- Acesse os recursos personalizados que são implantados e executados no Amazon Elastic Compute Cloud (Amazon EC2).

O Amazon GameLift oferece suporte aos seguintes métodos para estabelecer acesso:

- [Acesse recursos AWS com um perfil do IAM](#)
- [Acesse os recursos AWS com o emparelhamento de VPC](#)

Acesse recursos AWS com um perfil do IAM

Use uma perfil do IAM para especificar quem pode acessar seus recursos e definir limites para esse acesso. Partes confiáveis podem “assumir” uma função e obter credenciais de segurança temporárias que as autorizem a interagir com os recursos. Quando as partes fazem solicitações de API relacionadas ao recurso, elas devem incluir as credenciais.

Para configurar o acesso controlado por um perfil do IAM, execute as seguintes tarefas:

1. [Crie o perfil do IAM](#)

2. [Modifique os aplicativos para adquirir credenciais](#)
3. [Associe uma frota ao perfil do IAM](#)

Crie o perfil do IAM

Nesta etapa, você cria um perfil do IAM, com um conjunto de permissões para controlar o acesso aos seus recursos AWS e uma política de confiança que concede ao Amazon GameLift direitos de usar as permissões da função.

Para obter instruções sobre como configurar o perfil do IAM, consulte [Configurar uma função de serviço do IAM para a Amazon GameLift](#). Ao criar a política de permissões, escolha serviços, recursos e ações específicos com os quais seus aplicativos precisam trabalhar. Como melhor prática, limite o escopo das permissões o máximo possível.

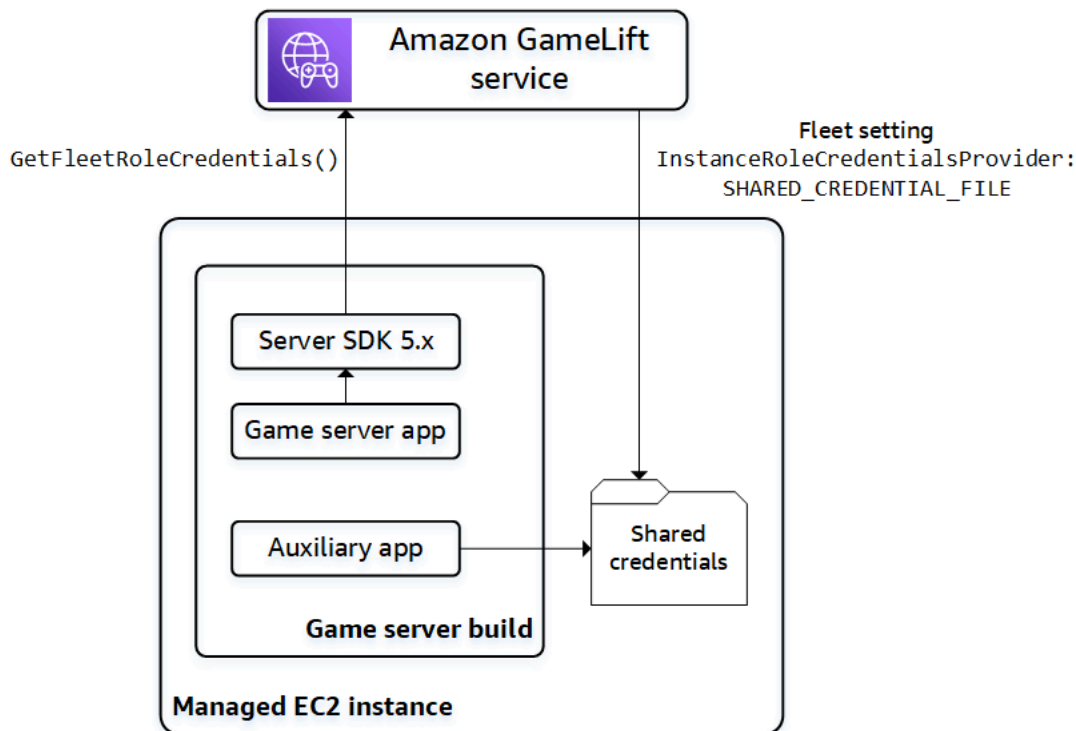
Depois de criar a função, anote o nome do recurso da Amazon (ARN) da função. Você precisa da função do ARN durante a criação da frota.

Modifique os aplicativos para adquirir credenciais

Nesta etapa, você configura seus aplicativos para adquirir credenciais de segurança para o perfil do IAM e usá-las ao interagir com seus recursos AWS. Consulte a tabela a seguir para determinar como modificar seus aplicativos com base (1) no tipo de aplicativo e (2) na versão do SDK do servidor que seu jogo usa para se comunicar com o Amazon GameLift.

	Aplicativos de servidor de jogos	Outros aplicativos
Usando o SDK do servidor versão 5.x	Chame o método SDK do servidor <code>GetFleetRoleCredentials()</code> a partir do código do servidor do jogo.	Adicione código ao aplicativo para extrair credenciais de um arquivo compartilhado na instância da frota.
Usando o SDK do servidor versão 4 ou anterior	Chame AWS Security Token Service (AWS STS) AssumeRole com a função do ARN.	Chame AWS Security Token Service (AWS STS) AssumeRole com a função do ARN.

Para jogos integrados ao SDK 5.x do servidor, esse diagrama ilustra como os aplicativos em sua versão de jogo implantada podem adquirir credenciais para o perfil do IAM.



Chamada `GetFleetRoleCredentials()` (SDK 5.x do servidor)

No código do seu servidor de jogos, que já deve estar integrado ao SDK 5.x do servidor do Amazon GameLift, chame `GetFleetRoleCredentials` ([C++](#)) ([C#](#)) ([Unreal](#)) para recuperar um conjunto de credenciais temporárias. Quando as credenciais expirarem, você poderá atualizá-las com outra chamada para `GetFleetRoleCredentials`.

Use credenciais compartilhadas (SDK 5.x do servidor)

Para aplicativos que não são de servidor que são implantados com compilações de servidores de jogos usando o SDK 5.x do servidor, adicione código para obter e usar credenciais armazenadas em um arquivo compartilhado. O Amazon GameLift gera um perfil de credenciais para cada instância da frota. As credenciais estão disponíveis para uso por todos os aplicativos na instância. O Amazon GameLift atualiza continuamente as credenciais temporárias.

Você deve configurar uma frota para gerar o arquivo de credenciais compartilhadas na criação da frota.

Em cada aplicativo que precisa usar o arquivo de credenciais compartilhado, especifique a localização do arquivo e o nome do perfil, da seguinte forma:

Windows:

```
[credentials]
shared_credential_profile= "FleetRoleCredentials"
shared_credential_file= "C:\\Credentials\\credentials"
```

Linux

```
[credentials]
shared_credential_profile= "FleetRoleCredentials"
shared_credential_file= "/local/credentials/credentials"
```

Exemplo: configurar um atendente do CloudWatch para coletar métricas para instâncias da frota do Amazon GameLift

Se você quiser usar um atendente do Amazon CloudWatch para coletar métricas, registros e rastreamentos de suas frotas do Amazon GameLift, use esse método para autorizar o agente a emitir os dados para sua conta. Nesse cenário, execute as seguintes etapas:

1. Recupere ou grave o arquivo `config.json` do atendente do CloudWatch.
2. Atualize o arquivo `common-config.toml` para que o atendente identifique o nome do arquivo de credenciais e o nome do perfil, conforme descrito acima.
3. Configure seu script de instalação de compilação do servidor de jogos para instalar e iniciar o atendente do CloudWatch.

Uso `AssumeRole()` (SDK do servidor 4)

Adicione código aos seus aplicativos para assumir o perfil do IAM e obter credenciais para interagir com seus recursos AWS. Qualquer aplicativo executado em uma instância de frota do Amazon GameLift com SDK de servidor 4 ou anterior pode assumir o perfil do IAM.

No código do aplicativo, antes de acessar um recurso AWS, o aplicativo deve chamar a operação da API AWS Security Token Service (AWS STS) [AssumeRole](#) e especificar o ARN da função. Essa operação retorna um conjunto de credenciais temporárias que autoriza o aplicação a acessar o recurso AWS. Para obter mais informações, consulte [Usar credenciais temporárias com recursos da AWS](#) no Guia do usuário do IAM.

Associe uma frota ao perfil do IAM

Depois de criar o perfil do IAM e atualizar os aplicativos na versão do seu servidor de jogos para obter e usar as credenciais de acesso, você pode implantar uma frota. Quando você configura a nova frota, defina os seguintes parâmetros:

- [instanceroLearn](#) – defina esse parâmetro como o ARN do perfil do IAM.
- [InstanceRoleCredentialsProvider](#) – Para solicitar que o Amazon GameLift gere um arquivo de credenciais compartilhado para cada instância da frota, defina esse parâmetro como SHARED_CREDENTIAL_FILE.

Você deve definir esses valores ao criar a frota. Eles não podem ser atualizados mais tarde.

Acesse os recursos AWS com o emparelhamento de VPC

Você pode usar o emparelhamento da Amazon Virtual Private Cloud (Amazon VPC) para se comunicar entre aplicações executadas em uma instância do Amazon GameLift e outro recurso AWS. Uma VPC é uma rede privada virtual definida por você e que inclui um conjunto de recursos gerenciados por meio da sua conta da Conta da AWS. Cada frota do Amazon GameLift tem sua própria VPC. Com o emparelhamento de VPCs, você pode estabelecer uma conexão de rede direta entre a VPC de sua frota e outros recursos da AWS.

O Amazon GameLift simplifica o processo de configuração de conexões de emparelhamento da VPC para seus servidores de jogos. Ele lida com solicitações de emparelhamento, atualiza tabelas de rota e configura as conexões, conforme necessário. Para obter instruções sobre como configurar o emparelhamento da VPC para os servidores de jogos, consulte [Emparelhamento de VPC para o Amazon GameLift](#).

Integre o cliente de jogo com o Amazon GameLift

Os tópicos nesta seção descrevem a funcionalidade gerenciada do Amazon GameLift, a qual você poderá adicionar a um serviço de back-end. Um serviço de back-end executa as seguintes tarefas:

- Solicita informações sobre sessões de jogos ativas do Amazon GameLift.
- Adiciona um jogador a uma sessão de jogo existente.
- Cria uma sessão de jogo e adiciona jogadores a ela.
- Altera metadados para uma sessão de jogo existente.

Para obter mais informações sobre como os clientes de jogos interagem com o Amazon GameLift e os servidores de jogos executados no Amazon GameLift, consulte [Amazon GameLift e interações entre cliente e servidor de jogos](#).

Pré-requisitos

- Uma Conta da AWS.
- Uma versão de servidor de jogo carregada no Amazon GameLift.
- Uma frota para hospedar os jogos.

Tópicos

- [Adicione GameLift a Amazon ao seu cliente de jogos](#)
- [Gerar IDs de jogador](#)

Adicione GameLift a Amazon ao seu cliente de jogos

Integre a Amazon GameLift aos componentes do jogo que precisam de informações sobre a sessão do jogo, crie novas sessões de jogo e adicione jogadores aos jogos. Dependendo da arquitetura do jogo, essa funcionalidade está serviços de back-end que processam tarefas como autenticação do jogador, marcação de jogos ou colocação da sessão de jogo.

Note

Para obter informações detalhadas sobre como configurar matchmaking para seu jogo GameLift hospedado na Amazon, consulte o [Guia do GameLift FlexMatch Desenvolvedor da Amazon](#).

Configure a Amazon GameLift em um serviço de back-end

Adicione código para inicializar um GameLift cliente Amazon e armazenar as configurações da chave. Esse código deve ser executado antes de qualquer código dependente da Amazon GameLift.

1. Defina uma configuração de cliente. Use a configuração padrão do cliente ou crie um objeto de configuração do cliente personalizado. Para obter mais informações, consulte [AWS::Client::ClientConfiguration\(C++\)](#) ou [AmazonGameLiftConfig\(C#\)](#).

A configuração do cliente especifica uma região de destino e um endpoint a serem usados ao entrar em contato com a Amazon. GameLift A região identifica o conjunto de recursos implantados (frotas, filas e marcador de jogos) a serem usados. A configuração do cliente padrão define a localização na região Leste dos EUA (Norte da Virgínia). Para usar qualquer outra Região, crie uma configuração personalizada.

2. Inicialize um GameLift cliente da Amazon. Use [Aws::GameLift:: GameLiftClient \(\)](#) (C++) ou [AmazonGameLiftClient\(\)](#) (C#) com uma configuração de cliente padrão ou uma configuração de cliente personalizada.
3. Adicione um mecanismo para gerar um identificador exclusivo para cada jogador. Para ter mais informações, consulte [Gerar IDs de jogador](#).
4. Colete e armazene informações a seguir:
 - Frota alvo — Muitas solicitações de GameLift API da Amazon devem especificar uma frota. Para fazer isso, use um ID de frota ou alias que aponte para a frota de destino. Como melhor prática, use os aliases de frota para poder trocar jogadores de uma frota para outra sem precisar atualizar os serviços de back-end.
 - Fila de destino – Para jogos que usam filas de várias frotas para colocar novas sessões de jogo, especifique o nome da fila a ser usada.
 - AWS credenciais — Todas as chamadas para a Amazon GameLift devem fornecer credenciais para quem hospeda Conta da AWS o jogo. Você adquire essas credenciais criando um usuário jogador, conforme descrito em [Configurar o acesso programático para o jogo](#). Dependendo de como você gerencia o acesso do usuário player, faça o seguinte:
 - Se você usa uma função para gerenciar as permissões de usuário do jogador, adicione um código para assumir a função antes de chamar uma GameLift API da Amazon. A solicitação para assumir a função retorna um conjunto de credenciais de segurança temporárias. Para obter mais informações, consulte [Mudar para uma função do IAM \(AWS API\)](#) no Guia do usuário do IAM.
 - Caso tenha credenciais de segurança de longo prazo, configure seu código para localizar e use as credenciais armazenadas. Consulte [Autenticar usando credenciais de longo prazo](#) no Guia de Referência de Ferramentas e SDKs da AWS . Para obter informações sobre como armazenar credenciais, consulte as referências de AWS API para [\(C++\)](#) e [\(.NET\)](#).
 - Caso tenha credenciais de segurança temporárias, adicione código para atualizar regularmente as credenciais usando o AWS Security Token Service (AWS STS), conforme descrito em [Usar credenciais de segurança temporárias com os SDKs da AWS](#) no Guia do usuário do IAM. O código deve solicitar novas credenciais antes que as antigas expirem.

Obter sessões de jogos

Adicione o código para descobrir as sessões de jogos disponíveis e gerenciar as configurações de sessões de jogos e metadados.

Procure sessões de jogos ativas

Use [SearchGameSessions](#) para obter informações sobre uma sessão de jogo específica, todas as sessões ativas ou sessões que atendem a um conjunto de critérios de pesquisa. Essa chamada retorna um [GameSession](#) objeto para cada sessão de jogo ativa que corresponde à sua solicitação de pesquisa.

Use critérios de pesquisa para obter uma lista filtrada de sessões de jogos ativas para jogadores ingressarem. Por exemplo, você pode filtrar sessões da seguinte maneira:

- Exclua sessões de jogos que sejam completas: `CurrentPlayerSessionCount = MaximumPlayerSessionCount`.
- Escolha sessões de jogos no tempo em que a sessão esteve em execução: avalie `CreationTime`.
- Encontre sessões de jogos baseadas em uma propriedade personalizada do jogo: `gameSessionProperties.gameMode = "brawl"`.

Gerencie sessões de jogos

Use qualquer uma das operações a seguir para recuperar ou atualizar as informações sobre a sessão do jogo.

- [DescribeGameSessionDetails\(\)](#) — Obtenha o status de proteção de uma sessão de jogo, além das informações da sessão de jogo.
- [UpdateGameSession\(\)](#) — Altere os metadados e as configurações de uma sessão de jogo conforme necessário.
- [GetGameSessionLogUrl](#) — Acesse os registros armazenados da sessão do jogo.

Criar sessões de jogos

Adicione um código para iniciar novas sessões de jogos nas frotas implantadas e disponibilizá-las para jogadores. Há duas opções para criar sessões de jogo, dependendo se você está implantando seu jogo em várias regiões da AWS ou em uma única região.

Crie uma sessão de jogo em uma fila com vários locais

Use [StartGameSessionPlacement](#) para fazer uma solicitação para uma nova sessão de jogo em uma fila. Para usar essa operação, crie uma fila. Isso determina onde a Amazon GameLift coloca a nova sessão de jogo. Para obter informações mais sobre filas e como usá-las, consulte [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).

Ao criar uma colocação de sessão do jogo, especifique o nome da fila a ser usada, um nome da sessão do jogo, um número máximo de jogadores simultâneos e um conjunto opcional de propriedades do jogo. Opcionalmente, também é possível fornecer uma lista de jogadores para ingressar automaticamente na sessão do jogo. Se você incluir dados de latência do jogador para regiões relevantes, a Amazon GameLift usará essas informações para colocar a nova sessão de jogo em uma frota que fornece a experiência de jogo ideal para os jogadores.

A colocação da sessão do jogo é um processo assíncrono. Após ter feito uma solicitação, você poderá deixar que ela seja bem-sucedida ou expire. Você também pode cancelar a solicitação a qualquer momento usando [StopGameSessionPlacement](#). Para verificar o status da sua solicitação de colocação, ligue [DescribeGameSessionPlacement](#).

Crie uma sessão do jogo em uma frota específica

Use [CreateGameSession](#) para criar uma nova sessão em uma frota especificada. Essa operação síncrona pode ser bem-sucedida ou falhar, dependendo dos recursos que a frota tem à disposição para hospedar uma nova sessão do jogo. Depois que a Amazon GameLift criar a nova sessão de jogo e devolver um [GameSession](#) objeto, você poderá juntar jogadores a ela.

Ao usar esta operação, forneça um ID de frota ou alias, um nome de sessão e um número máximo de jogadores simultâneos para o jogo. Você também pode incluir um conjunto de propriedades do jogo. As propriedades do jogo são definidas em uma matriz de pares de chave-valor.

Se você usar o GameLift recurso de proteção de recursos da Amazon para limitar o número de sessões de jogo que um jogador pode criar, forneça o ID de jogador do criador da sessão de jogo.

Adicionar um jogador a uma sessão de jogo

Adicione código para reservar um slot de jogador em uma sessão de jogo ativa e conecte clientes de jogos a sessões de jogos.

1. Reserve um slot de jogador em uma sessão de jogo

Para reservar um slot de jogador, crie uma nova sessão de jogador para a sessão de jogo. Para obter mais informações sobre sessões de jogadores, consulte [Como os jogadores se conectam aos jogos](#).

Há duas maneiras de criar novas sessões de jogador:

- Use [StartGameSessionPlacement](#) para reservar vagas para um ou mais jogadores na nova sessão de jogo.
- Reserve vagas de jogador para um ou mais jogadores usando [CreatePlayerSession](#) ou [CreatePlayerSessions](#) com um ID de sessão de jogo.

A Amazon GameLift primeiro verifica se a sessão do jogo está aceitando novos jogadores e tem vagas disponíveis para jogadores. Se for bem-sucedido, a Amazon GameLift reserva um espaço para o jogador, cria a nova sessão de jogador e retorna um [PlayerSession](#) objeto. Esse objeto contém o nome DNS, o endereço IP e a porta que um cliente de jogo precisa para se conectar à sessão do jogo.

Uma solicitação de sessão de jogador deve incluir um ID exclusivo para cada jogador. Para ter mais informações, consulte [Gerar IDs de jogador](#).

Uma sessão de jogador pode incluir um conjunto de dados de jogador personalizado. Esses dados são armazenados no objeto de sessão do player recém-criado, que você pode recuperar chamando [DescribePlayerSessions\(\)](#). A Amazon GameLift também passa esse objeto para o servidor do jogo quando o jogador se conecta diretamente à sessão do jogo. Ao solicitar várias sessões de jogadores, forneça uma sequência de dados do jogador para cada jogador mapeado para o ID do jogador na solicitação.

2. Conecte-se a uma sessão do jogo

Adicione código ao cliente do jogo para recuperar o objeto de `PlayerSession`, que contém as informações de conexão da sessão do jogo. Use essas informações para estabelecer uma conexão direta com o servidor.

- Conecte-se usando a porta especificada e o nome DNS ou endereço IP atribuído ao processo do servidor.
- Se a geração de certificados TLS estiver habilitada para suas frotas, deverá se conectar usando o nome DNS e a porta.

- Se o seu servidor de jogo valida as conexões de entrada dos jogadores, faça referência ao ID da sessão do jogador.

Depois de fazer a conexão, o processo do cliente e do servidor do jogo se comunicam diretamente sem envolver a Amazon GameLift. O servidor mantém comunicação com GameLift a Amazon para relatar o status da conexão do jogador, o status de saúde e muito mais. Se o servidor do jogo valida os novos jogadores, ele verifica se o ID da sessão do jogador corresponde a um espaço reservado na sessão do jogo e aceita ou nega a conexão do jogador. Quando o jogador se desconectar, o processo do servidor informará a conexão desfeita.

Use as propriedades da sessão do jogo

Seu cliente de jogo pode passar dados para uma sessão de jogo usando uma propriedade do jogo. As propriedades do jogo são pares de valores-chave que seu servidor de jogo pode adicionar, ler, listar e alterar. Você pode transmitir uma propriedade do jogo ao criar uma nova sessão de jogo ou posteriormente quando a sessão do jogo estiver ativa. Uma sessão de jogo pode conter até 16 propriedades do jogo. Você não pode excluir as propriedades do jogo.

Por exemplo, seu jogo oferece os seguintes níveis de dificuldade: Novice EasyIntermediate,, Expert e. Um jogador escolhe eEasy, em seguida, começa o jogo. Seu cliente de jogo solicita uma nova sessão de jogo da Amazon GameLift usando uma `StartGameSessionPlacement` ou `CreateGameSession` conforme explicado nas seções anteriores. Na solicitação, o cliente passa isso:{"Key": "Difficulty", "Value":"Easy"}.

Em resposta à solicitação, a Amazon GameLift cria um `GameSession` objeto que contém a propriedade do jogo especificada. A Amazon GameLift então instrui um servidor de jogo disponível a iniciar a nova sessão de jogo e passar o `GameSession` objeto. O servidor do jogo inicia uma sessão de jogo com um `Difficulty` deEasy.

Saiba mais

- [GameProperty tipo de dados](#)
- [SearchGameSessions\(\) exemplos](#)
- [UpdateGameSession\(\) GameProperties parâmetro](#)

Gerar IDs de jogador

O Amazon GameLift usa uma sessão de jogador para representar um jogador conectado a uma sessão de jogo. O Amazon GameLift cria uma sessão de jogador sempre que um jogador se conecta a uma sessão de jogo usando um cliente de jogo integrado ao Amazon GameLift. Quando um jogador sai de um jogo, a sessão do jogador termina. O Amazon GameLift não reutiliza sessões de jogadores.

O exemplo de código a seguir gera aleatoriamente IDs de jogadores exclusivos:

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

Para obter mais informações sobre sessões de jogadores, consulte [Visualizar dados nas sessões de jogo e jogador](#).

Mecanismos de jogo e Amazon GameLift

Você pode usar o serviço gerenciado do Amazon GameLift com a maioria dos principais mecanismos de jogos que oferecem suporte a bibliotecas C++ ou C#, inclusive O3DE, Unreal Engine e Unity. Compile a versão que você precisa para o seu jogo; consulte os arquivos LEIA-ME de cada versão para mais instruções de compilação e requisitos mínimos. Para obter mais informações sobre os SDKs do Amazon GameLift disponíveis e as plataformas de desenvolvimento e os sistemas operacionais compatíveis, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#) para servidores de jogos.

Além das informações específicas do mecanismo fornecidas neste tópico, encontre ajuda adicional para integração do Amazon GameLift aos servidores de jogos, clientes e serviços nos seguintes tópicos:

- [Roteiro de hospedagem gerenciada do Amazon GameLift](#) – Um fluxo de trabalho em seis etapas para integrar com êxito o Amazon GameLift ao jogo e configurar recursos de hospedagem.
- [Adicione o Amazon GameLift ao servidor de jogos](#) – Instruções detalhadas sobre como integrar o Amazon GameLift a um servidor de jogos.
- [Adicione GameLift a Amazon ao seu cliente de jogos](#) – Instruções detalhadas sobre a integração a um cliente ou serviço de jogo, inclusive a criação de sessões de jogos e a entrada de jogadores em jogos.

O3DE

Servidores de jogos

Prepare os servidores de jogos para hospedagem no Amazon GameLift usando o [Server SDK do Amazon GameLift para C++](#). Consulte [Adicione o Amazon GameLift ao servidor de jogos](#) para obter ajuda na integração da funcionalidade necessária ao servidor de jogos.

Clientes e serviços de jogos

Ative os clientes e/ou serviços de jogos para interagir com o serviço do Amazon GameLift, por exemplo, para encontrar sessões de jogos disponíveis ou criar sessões e adicionar jogadores a jogos. A principal funcionalidade do cliente é fornecida no [AWS SDK for C++](#). Para integrar o Amazon GameLift ao seu projeto de jogo O3DE, consulte [Adicione o Amazon GameLift a um cliente e servidor de jogos O3DE](#) e [Adicione GameLift a Amazon ao seu cliente de jogos](#).

Unreal Engine

Servidores de jogos

Prepare os servidores de jogos para hospedagem no Amazon GameLift adicionando o [Server SDK do Amazon GameLift para Unreal Engine](#) ao projeto e implementando a funcionalidade de servidor necessária. Para ajudar a configurar o plug-in Unreal Engine e adicionar o código do Amazon GameLift, consulte [Integre a Amazon GameLift em um projeto Unreal Engine](#).

Clientes e serviços de jogos

Ative os clientes e/ou serviços de jogos para interagir com o serviço do Amazon GameLift, por exemplo, para encontrar sessões de jogos disponíveis ou criar sessões e adicionar jogadores a jogos. A principal funcionalidade do cliente é fornecida no [AWS SDK for C++](#). Para integrar o Amazon GameLift ao projeto de jogo do Unreal Engine, consulte [Adicione GameLift a Amazon ao seu cliente de jogos](#).

Unity

Servidores de jogos

Prepare os servidores de jogos para hospedagem no Amazon GameLift adicionando o [Server SDK do Amazon GameLift para C#](#) ao projeto e implementando a funcionalidade de servidor necessária.

Para ajudar a configurar o Unity e adicionar o código do Amazon GameLift, consulte [Integre o Amazon GameLift em um projeto do Unity](#).

Cientes e serviços de jogos

Ative os clientes e/ou serviços de jogos para interagir com o serviço do Amazon GameLift, por exemplo, para encontrar sessões de jogos disponíveis ou criar sessões e adicionar jogadores a jogos. A principal funcionalidade do cliente é fornecida no [AWS SDK for .NET](#). Para integrar o Amazon GameLift ao seu projeto de jogo do Unity, consulte [Adicione GameLift a Amazon ao seu cliente de jogos](#).

Outros mecanismos

Para obter uma lista completa dos SDKs do Amazon GameLift disponíveis para servidores de jogos e clientes, consulte [the section called “Suporte ao desenvolvimento”](#).

Adicione o Amazon GameLift a um cliente e servidor de jogos O3DE

Você pode usar o O3DE, um mecanismo 3D de código aberto, multiplataforma e em tempo real para criar experiências interativas de alto desempenho, incluindo jogos e simulações. O renderizador e as ferramentas O3DE são agrupados em uma estrutura modular que você pode modificar e estender com suas ferramentas de desenvolvimento preferidas.

A estrutura modular usa Gems que contêm bibliotecas com interfaces e ativos padrão. Selecione seus próprios Gems para escolher quais funcionalidades adicionar com base em seus requisitos.

O Amazon GameLift Gem fornece os seguintes recursos:

Integração com o Amazon GameLift

Uma estrutura para estender a camada de rede O3DE e permitir que o Multiplayer Gem funcione com a solução de servidor dedicado do Amazon GameLift. O Gem fornece integrações com o [SDK do servidor Amazon GameLift](#) e com o cliente do SDK da AWS (para chamar o próprio serviço Amazon GameLift).

Gerenciamento de compilação e pacotes

Instruções para empacotar e, opcionalmente, carregar a versão do servidor dedicado e um aplicativo AWS Cloud Development Kit (AWS CDK) (AWS CDK) para configurar e atualizar recursos.

Configuração do Amazon GameLift Gem

Siga os procedimentos nesta seção para configurar o Amazon GameLift Gem no O3DE.

Pré-requisitos

- Configure sua conta da AWS no Amazon GameLift. Para obter mais informações, consulte [Configurar um Conta da AWS](#).
- Configure as credenciais da AWS para o O3DE. Para obter mais informações, consulte [Configuração das credenciais da AWS](#).
- Configure o AWS CLI e AWS CDK. Para obter mais informações, [AWS Command Line Interface](#) e [AWS Cloud Development Kit \(AWS CDK\)](#).

Ative o Amazon GameLift Gem e suas dependências

1. Abra o Gerente de projetos.
2. Abra o menu abaixo do seu projeto e escolha Editar configuração do projeto....
3. Selecione a opção Configurar o Gems.
4. Ative o Amazon GameLift Gem e os seguintes Gems dependentes:
 - [Core principal da AWS](#) – Forneça a estrutura a ser usada para os Serviços da AWS no O3DE.
 - [Multiplayer Gem](#) – Fornece funcionalidade multijogador ao estender a estrutura de rede.

Inclua a biblioteca estática Amazon GameLift Gem

1. Inclua o `Gem::AWSGameLift.Server.Static` como `BUILD_DEPENDENCIES` para o destino do seu servidor de projeto.

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

2. `AWSServiceDef` como necessário para o componente do sistema do servidor de projetos.

```
void
YourProjectServerSystemComponent::GetRequiredServices(AZ::ComponentDescriptor::DependencyA
required)
{
    ...
    required.push_back(AZ_CRC_CE("AWSServiceDef"));
    ...
}
```

3. (Opcional) Para fazer solicitações de serviço do Amazon GameLift em C++, `Gem::AWSServiceDef.Client.Static` inclua no `BUILD_DEPENDENCIES` para seu cliente de destino.

```
ly_add_target(
    NAME YourProject.Client.Static STATIC
    ...
    BUILD_DEPENDENCIES
    PUBLIC
    ...
    PRIVATE
    ...
    Gem::AWSServiceDef.Client.Static
    Gem::AWSServiceDef.Client.Static
}
```

Integre seu jogo e servidor dedicado

Gerencie sessões de jogo em seu jogo e em um servidor de jogo dedicado com a [Integração de gerenciamento de sessões](#). [Para oferecer suporte ao FlexMatch, consulte Integração com o FlexMatch](#).

Integre a Amazon GameLift em um projeto Unreal Engine

Este tópico explica como configurar o plug-in SDK do servidor Amazon GameLift C++ para o Unreal Engine e integrá-lo aos seus projetos de jogos.

Recursos adicionais:

- [Plug-in do SDK de servidor para site de download Unreal](#)
- [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)
- [the section called “Suporte ao desenvolvimento”](#)

Pré-requisitos

Antes de prosseguir, garanta revisado os seguintes pré-requisitos:

Pré-requisitos

- Um computador capaz de executar o Unreal Engine. Para obter mais informações sobre os requisitos do Unreal Engine, consulte a documentação de [Especificações de hardware e software do Unreal Engine](#).
- Microsoft Visual Studio 2019 ou versão mais recente.
- CMake versão 3.1 ou posterior.
- Python versão 3.6 ou posterior.
- Um cliente Git disponível no PATH.
- Uma conta da Epic Games. Crie uma conta no site oficial da [Unreal Engine](#).
- Uma GitHub conta associada à sua conta do Unreal Engine. Para obter mais informações, consulte [Acessando o código-fonte do Unreal Engine GitHub no](#) site do Unreal Engine.

Note

GameLift Atualmente, a Amazon oferece suporte às seguintes versões do Unreal Engine:

- 4.22
- 4.23
- 4,24
- 4,25
- 4.26
- 4,27
- 5.1.0
- 5.1.1
- 5.2

- 5.3

Crie o Unreal Engine a partir da fonte

As versões padrão do editor do Unreal Engine, baixadas por meio do inicializador do Epic, permitem apenas compilações de aplicativos clientes do Unreal. Para criar um aplicativo de servidor do Unreal, você precisa baixar e criar o Unreal Engine a partir do código-fonte, usando o repositório no Github do Unreal Engine. Para obter mais informações, consulte o tutorial [Compilar o Unreal Engine a partir da origem](#) no site de documentação do Unreal Engine.

Note

Se você ainda não fez isso, siga as instruções em [Acessando o código-fonte do Unreal Engine GitHub](#) para vincular sua GitHub conta à sua conta da Epic Games.

Para clonar a fonte do Unreal Engine em seu ambiente de desenvolvimento

1. Clone a fonte do Unreal Engine em seu ambiente de desenvolvimento em uma ramificação de sua escolha.

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

2. Confira a tag da versão que você está usando para desenvolver seu jogo. Por exemplo, o exemplo a seguir confere a versão 5.1.1 do Unreal Engine:

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

3. Navegue até a pasta raiz do repositório local. Quando você estiver na pasta raiz, execute o seguinte arquivo: `Setup.bat`.
4. Enquanto estiver na pasta raiz, execute também o arquivo: `GenerateProjectFiles.bat`.
5. Depois de executar os arquivos das etapas anteriores, um arquivo de solução do Unreal Engine, `UE5.sln`, é criado. Abra o Visual Studio e, no editor do Visual Studio, abra o arquivo `UE5.sln`.
6. No Visual Studio, abra o menu Exibir e escolha a opção Gerenciador de Soluções. Isso abre o menu de contexto do nó do projeto Unreal. Na janela Gerenciador de Soluções, clique com o botão direito do mouse no arquivo `UE5.sln` (ele pode ser listado apenas como UE5) e escolha Compilação para criar o projeto do Unreal com o destino Win64 do Editor de Desenvolvimento.

Note

A construção pode levar mais de uma hora para ser concluída.

Quando a compilação estiver concluída, você estará pronto para abrir o Editor de Desenvolvimento do Unreal e criar ou importar um projeto.

Configure seu projeto do Unreal para o plug-in

Siga estas etapas para preparar o plug-in Amazon GameLift Server SDK para Unreal Engine para seus projetos de servidor de jogos.

Para configurar seu projeto para o plug-in

1. Com o Visual Studio aberto, navegue até o painel Gerenciador de Soluções e escolha o arquivo UE5 para abrir o menu de contexto do projeto do Unreal. No menu de contexto, escolha a opção Definir como projeto de Startup.
2. Na parte superior da janela do Visual Studio, escolha Iniciar depuração (seta verde).

Essa ação inicia sua nova instância de código-fonte do Unreal Editor. Para obter mais informações sobre o uso do Unreal Editor, consulte [Interface do Unreal Editor](#) no site de documentação do Unreal Engine.

3. Feche a janela do Visual Studio que você abriu, pois o Unreal Editor abre outra janela do Visual Studio que contém o projeto do Unreal e seu projeto de jogo.
4. No editor do Unreal, siga um destes procedimentos:
 - Escolha um projeto Unreal existente que você deseja integrar com a Amazon GameLift.
 - Criar um novo projeto da . Para experimentar o GameLift plug-in da Amazon para Unreal, tente usar o modelo de terceira pessoa do motor Unreal. Para obter mais informações sobre esse modelo, consulte [Modelo de terceira pessoa](#) no site de documentação do Unreal Engine.

Como alternativa, configure um novo projeto com as seguintes configurações:

- C++
- Com conteúdo inicial
- Área de Trabalho

- Um nome de projeto. Nos exemplos deste tópico, nomeamos nosso projeto `GameLiftUnrealApp`.
5. No Gerenciador de Soluções do Visual Studio, navegue até o local do seu projeto do Unreal. Na pasta Source do Unreal, encontre um arquivo chamado *Your-application-name*.Target.cs.

Por exemplo: `GameLiftUnrealApp.Target.cs`.

6. Faça uma cópia deste arquivo e nomeie a cópia: *Your-application-name*Server.Target.cs.
7. Abra o novo arquivo e faça as seguintes alterações:
 - Altere `class` e `constructor` para corresponder ao nome do arquivo.
 - Altere o `Type` formulário `TargetType.Game` para `TargetType.Server`.
 - O arquivo final será semelhante ao seguinte exemplo:

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

Seu projeto agora está configurado para aceitar o plug-in Amazon GameLift Server SDK.

A próxima tarefa é criar as bibliotecas do SDK do servidor C++ para Unreal para que você possa importá-las para o seu projeto.

Para construir as bibliotecas SDK do servidor C++ para o Unreal

1. Baixe o [plug-in Amazon GameLift C++ Server SDK para Unreal](#).

Note

Colocar o SDK no diretório de download padrão pode resultar em falha de compilação devido ao caminho exceder o limite de 260 caracteres. Por exemplo: `C:\Users\Administrator\Downloads\GameLift-SDK-Release-06_15_2023\GameLift-Cpp-ServerSDK-5.0.4`
Recomendamos que você mova o SDK para outro diretório, por exemplo `C:\GameLift-Cpp-ServerSDK-5.0.4`.

2. Faça download do OpenSSL e instale-o. Para obter mais informações sobre como baixar o OpenSSL, leia a documentação de [compilação e instalação do OpenSSL](#) do Github.

Para obter mais informações, leia a documentação do OpenSSL [Notas para plataformas Windows](#).

Note

A versão do OpenSSL que você usa para criar o SDK do servidor GameLift Amazon deve corresponder à versão do OpenSSL usada pela Unreal para empacotar seu servidor de jogos. Você pode encontrar informações sobre a versão no diretório `...Engine\Source\ThirdParty\OpenSSL` de instalação do Unreal.

3. Com as bibliotecas baixadas, crie as bibliotecas do SDK do servidor C++ para Unreal Engine.

No diretório `GameLift-Cpp-ServerSDK-<version>` do SDK baixado, compile com o parâmetro `-DBUILD_FOR_UNREAL=1` e crie o SDK do servidor. Os exemplos a seguir mostram como compilar usando `cmake`.

Execute os seguintes comandos em seu terminal:

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

A compilação do Windows cria os seguintes arquivos binários na pasta `out\gamelift-server-sdk\Release`:

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

Copie os dois arquivos da biblioteca para a `ThirdParty\GameLiftServerSDK\Win64` pasta no pacote de plug-ins Amazon GameLift Unreal Engine.

Use o procedimento a seguir para importar o GameLift plug-in da Amazon para seu projeto de exemplo.

Importe o GameLift plug-in da Amazon

1. Localize a `GameLiftServerSDK` pasta que você extraiu do plug-in no procedimento anterior.
2. Localize o `Plugins` na pasta raiz do seu projeto de jogo. (Se a pasta não existir, crie-a lá.)
3. Copie a `GameLiftServerSDK` pasta para `Plugins` o.

Isso permitirá que o projeto Unreal veja o plug-in.

4. Adicione o plug-in Amazon GameLift Server SDK ao `.uproject` arquivo do jogo.

No exemplo, o aplicativo é chamado `GameLiftUnrealApp`, então o arquivo será `GameLiftUnrealApp.uproject`.

5. Edite o arquivo `.uproject` para adicionar o plug-in ao seu projeto de jogo.

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

6. Certifique-se de que o `ModuleRules` jogo dependa do plug-in. Abra o `.Build.cs` arquivo e adicione a dependência do Amazon GameLiftServer SDK. Este arquivo pode ser encontrado no *`Your-application-name/Source//Your-application-name/`*.

Por exemplo, o caminho do arquivo do tutorial é `../GameLiftUnrealApp/Source/GameLiftUnrealApp/GameLiftUnrealApp.Build.cs`.

7. Adicione `"GameLiftServerSDK"` ao final da lista de `PublicDependencyModuleNames`.

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

O plug-in agora deve estar funcionando para seu aplicativo. Continue com a próxima seção para integrar a GameLift funcionalidade da Amazon ao seu jogo.

Adicione o código GameLift do servidor Amazon ao seu projeto Unreal

Você configurou e configurou seu ambiente Unreal Engine e agora pode integrar um servidor de jogos com a Amazon GameLift. O código apresentado neste tópico faz as chamadas obrigatórias para o GameLift serviço da Amazon. Ele também implementa um conjunto de funções de retorno de chamada que respondem às solicitações do serviço da Amazon GameLift . Para obter mais informações sobre cada função e o que o código faz, consulte [Inicializar o processo do servidor](#). Para obter mais informações sobre as ações e os tipos de dados do SDK usados nesse código, leia [Referência do SDK do servidor do Amazon GameLift para Unreal Engine](#).

Para inicializar um servidor de jogos com a Amazon GameLift, use o procedimento a seguir.

Note

O código GameLift específico da Amazon fornecido na seção a seguir depende do uso de um sinalizador de `WITH_GAMELIFT` pré-processador. Esse sinalizador é verdadeiro somente quando essas duas condições são atendidas:

- `Target.Type == TargetRules.TargetType.Server`
- Os plug-ins encontraram os binários GameLift do SDK do servidor Amazon.

Isso garante que somente as compilações do Unreal Server invoquem a API de back-end da GameLift Amazon. Também permite que você escreva um código que será executado corretamente para todos os diferentes alvos do Unreal que seu jogo possa produzir.

Integre um servidor de jogos com a Amazon GameLift

1. No Visual Studio, abra o arquivo `.sln` do aplicativo. No nosso exemplo, o arquivo `GameLiftUnrealApp.sln` é encontrado na pasta raiz.
2. Com a solução aberta, localize o arquivo `Your-application-nameGameMode.h` do seu aplicativo. Exemplo: `GameLiftUnrealAppGameMode.h`.
3. Altere o arquivo de cabeçalho para alinhá-lo com o código de exemplo a seguir. Certifique-se de substituir "GameLiftUnrealApp" pelo nome do seu próprio aplicativo.

```
#pragma once

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

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

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

public:
    AGameLiftUnrealAppGameMode();

protected:
    virtual void BeginPlay() override;

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

    void InitGameLift();
}
```



```
};
```

4. Abra o arquivo *Your-application-name*GameMode.cpp de origem relacionado. Em nosso Exemplo: GameLiftUnrealAppGameMode.cpp. e altere o código para alinhar com o código de exemplo a seguir. Certifique-se de substituir "GameLiftUnrealApp" pelo nome do seu próprio aplicativo.

Este exemplo mostra como adicionar todos os elementos necessários para a integração com a Amazon GameLift, conforme descrito em [Adicionar Amazon GameLift ao seu servidor de jogos](#). Isso inclui:

- Inicializando um cliente de GameLift API da Amazon.
- Implementação de funções de retorno de chamada para responder às solicitações do GameLift serviço da AmazonOnStartGameSession, incluindoOnProcessTerminate, e. onHealthCheck
- Chamando ProcessReady () com uma porta designada para notificar a Amazon GameLiftservice quando estiver pronta para sediar sessões de jogo.

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

#include "UObject/ConstructorHelpers.h"

DEFINE_LOG_CATEGORY(GameServerLog);

AGameLiftUnrealAppGameMode::AGameLiftUnrealAppGameMode()
{
    // set default pawn class to our Blueprinted character
    static ConstructorHelpers::FClassFinder<APawn> PlayerPawnBPClass(TEXT("/Game/ThirdPerson/Blueprints/BP_ThirdPersonCharacter"));
    if (PlayerPawnBPClass.Class != NULL)
    {
        DefaultPawnClass = PlayerPawnBPClass.Class;
    }
}

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

```
#endif
}

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

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

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

    //AuthToken returned from the "aws gamelift get-compute-auth-token" API. Note
    this will expire and require a new call to the API after 15 minutes.
    if (FParse::Value(FCommandLine::Get(), TEXT("-authtoken="),
serverParameters.m_authToken))
    {
        UE_LOG(GameServerLog, Log, TEXT("AUTH_TOKEN: %s"),
*serverParameters.m_authToken)
    }

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

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

    //The WebSocket URL (GameLiftServiceSdkEndpoint).
    if (FParse::Value(FCommandLine::Get(), TEXT("-websocketurl="),
serverParameters.m_webSocketUrl))
    {

```

```

        UE_LOG(GameServerLog, Log, TEXT("WEBSOCKET_URL: %s"),
*serverParameters.m_webSocketUrl)
    }

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

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

    //Implement callback function onStartGameSession
    //GameLift sends a game session activation request to the game server
    //and passes a game session object with game properties and other settings.
    //Here is where a game server takes action based on the game session object.
    //When the game server is ready to receive incoming player connections,
    //it invokes the server SDK call ActivateGameSession().
    auto onGameSession = [=](Aws::GameLift::Server::Model::GameSession gameSession)
    {
        FString gameSessionId = FString(gameSession.GetGameSessionId());
        UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"),
*gameSessionId);
        gameLiftSdkModule->ActivateGameSession();
    };

    m_params.OnStartGameSession.BindLambda(onGameSession);

    //Implement callback function OnProcessTerminate
    //GameLift invokes this callback before shutting down the instance hosting this
game server.
    //It gives the game server a chance to save its state, communicate with
services, etc.,
    //and initiate shut down. When the game server is ready to shut down, it
invokes the
    //server SDK call ProcessEnding() to tell GameLift it is shutting down.
    auto onProcessTerminate = [=]()
    {
        UE_LOG(GameServerLog, Log, TEXT("Game Server Process is terminating"));
        gameLiftSdkModule->ProcessEnding();
    };

```

```
m_params.OnTerminate.BindLambda(onProcessTerminate);

//Implement callback function OnHealthCheck
//GameLift invokes this callback approximately every 60 seconds.
//A game server might want to check the health of dependencies, etc.
//Then it returns health status true if healthy, false otherwise.
//The game server must respond within 60 seconds, or GameLift records 'false'.
//In this example, the game server always reports healthy.
auto onHealthCheck = []()
{
    UE_LOG(GameServerLog, Log, TEXT("Performing Health Check"));
    return true;
};

m_params.OnHealthCheck.BindLambda(onHealthCheck);

//The game server gets ready to report that it is ready to host game sessions
//and that it will listen on port 7777 for incoming player connections.
m_params.port = 7777;

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

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

5. Crie um projeto de jogo para os dois tipos de alvo a seguir: Editor de Desenvolvimento e Servidor de Desenvolvimento.

Note

Não é necessário reconstruir a solução. Em vez disso, crie apenas o projeto na pasta Games que corresponde ao nome do seu aplicativo. Caso contrário, o Visual Studio reconstrói todo o projeto UE5, o que pode levar até uma hora.

6. Quando as duas compilações estiverem concluídas, feche o Visual Studio e abra o arquivo `.uproject` do seu projeto para abri-lo no Unreal Editor.
7. No Unreal Editor, empacote a versão do servidor do seu jogo. Para escolher um destino, vá para Plataformas, Windows e selecione ***our-application-nameServidor Y***.
8. Para iniciar o processo de criação do aplicativo do servidor, acesse Plataformas, Windows e selecione Projeto de pacote. Quando a compilação estiver concluída, você deverá ter um executável. No caso do nosso exemplo, o nome do arquivo é `GameLiftUnrealAppServer.exe`.
9. A criação de um aplicativo de servidor no Unreal Editor produz dois executáveis. Um está localizado na raiz da pasta de compilação do jogo e atua como um invólucro para o executável real do servidor.

Ao criar uma GameLift frota da Amazon com a construção do seu servidor, recomendamos que você transmita o executável real do servidor como o caminho de lançamento da configuração de tempo de execução. Por exemplo, na pasta de criação do jogo, você pode ter um arquivo `GameLiftFPS.exe` na raiz e outro em `\GameLiftFPS\Binaries\Win64\GameLiftFPS\Server.exe`. Ao criar uma frota, recomendamos que você use `C:\GameLiftFPS\Binaries\Win64\GameLiftFPS\Server.exe` como o caminho de inicialização da configuração de runtime.

10. Certifique-se de abrir as portas UDP necessárias na GameLift frota da Amazon, para que o servidor do jogo possa se comunicar com os clientes do jogo. Por padrão, o Unreal Engine usa porta 7777. Para obter mais informações, consulte o guia [UpdateFleetPortSettings](#) de referência da API GameLift de serviços da Amazon.
11. Crie um arquivo `install.bat` para a versão do seu jogo. Esse script de instalação é executado sempre que a compilação do jogo é implantada em uma GameLift frota da Amazon. Veja um arquivo de exemplo `install.bat`:

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

Para algumas versões do Unreal Engine, o `install.bat` deveria ser

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

Note

O caminho do arquivo para o arquivo `<>PrereqSetup_x64.exe` é `Engine\Extras\Redist\en-us`.

12. Agora você pode empacotar e enviar sua versão do jogo para a Amazon GameLift.

A versão do OpenSSL que você empacota com a compilação do jogo precisa corresponder à versão que o mecanismo de jogo usou ao criar o servidor do jogo. Certifique-se de empacotar a versão correta do OpenSSL com a compilação do servidor do jogo. Para o sistema operacional Windows, o formato OpenSSL é `.dll`.

Note

Package as DLLs do OpenSSL na versão do seu servidor de jogos. Certifique-se de empacotar a mesma versão do OpenSSL que você usou ao criar o servidor do jogo.

- `libssl-1_1-x64.dll`

`libcrypto-1_1-x64.dll`

Package suas dependências junto com o executável do servidor de jogos na raiz de um arquivo zip. Por exemplo, as dlls `openssl-lib` devem estar no mesmo diretório do arquivo `.exe`.

Próximas etapas

Você configurou e configurou seu ambiente Unreal Engine e agora pode começar a integrar a Amazon GameLift ao seu jogo.

Para obter mais informações sobre como adicionar GameLift a Amazon ao seu jogo, consulte o seguinte:

- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Referência do SDK do servidor do Amazon GameLift para Unreal Engine](#)

Para obter instruções sobre como testar seu jogo, consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#) .

Integre o Amazon GameLift em um projeto do Unity

Este tópico explica como configurar o plug-in do Server SDK para C# do Amazon GameLift para Unity e integrá-lo aos seus projetos de jogos.

Recursos adicionais:

- [Site de download do SDK do servidor do Amazon GameLift](#)
- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)
- [the section called “Suporte ao desenvolvimento”](#)

Pré-requisitos

Para usar o plug-in do SDK do servidor C# do Amazon GameLift para Unity, você precisa dos seguintes componentes:

- Um ambiente de desenvolvimento e uma versão do Unity Editor que o plug-in suporta (consulte [Suporte ao desenvolvimento com a Amazon GameLift](#)). Para obter informações sobre as versões do Unity, consulte [Requisitos do sistema para Unity](#) na documentação do Unity.
- O plug-in SDK do servidor do Amazon GameLift para o pacote do Unity. Esse pacote inclui o SDK 5+ do servidor para C#. Você poderá baixar o pacote neste site: [Conceitos básicos do Amazon GameLift](#).
- O registro com escopo de terceiros do UnityNuGet. Essa ferramenta gerencia DLLs de terceiros. Para obter mais informações, consulte o repositório [UnityNuGet](#) no Github.

Configurar o UnityNuGet

Se você não tiver o UnityNuGet configurado para seu projeto de jogo, use as etapas a seguir para instalar a ferramenta usando o gerenciador de pacotes do Unity. Como alternativa, você pode usar a NuGet CLI para baixar manualmente as DLLs. Para obter mais informações, consulte do SDK do servidor C# do Amazon GameLift para Unity README.

Para integrar o UnityNuGet ao projeto de jogo do Unity.

1. Com seu projeto aberto no Unity Editor, acesse o menu principal e selecione Editar, Configurações do projeto. Nas opções, escolha a seção Gerenciador de pacotes e abra o grupo Registros com Escopo.
2. Escolha o botão + e insira os seguintes valores para o registro com escopo do UnityNuGet:

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

3. Para usuários da versão 2021 do Unity:

Depois de configurar o UnityNuGet, verifique se há erros `Assembly Version Validation` exibidos no console do Unity. Esses erros ocorrem se os redirecionamentos de vinculação para assemblies com nomes fortes nos pacotes do NuGet não estiverem sendo resolvidos corretamente para caminhos dentro do seu projeto do Unity. Para resolver esse problema, configure a validação da versão de montagem do Unity:

- a. No Unity Editor, vá para o menu principal e selecione Editar, Configurações do projeto e abra a seção Jogador.
- b. Desmarque a opção Validação da versão de montagem.

Instalar o plug-in

Use o procedimento a seguir para instalar o plug-in do SDK do servidor C# do Amazon GameLift para Unity e configurar o registro do log4net.

Para instalar o plugin

1. Com seu projeto aberto no Unity Editor, vá até o menu principal e selecione Window, Gerenciador de pacotes.
2. Escolha o botão + para adicionar um novo pacote. Escolha a opção Adicionar pacote do tarball.
3. Em Selecionar pacotes no disco, localize o plug-in do Server SDK para C# do Amazon GameLift para baixar arquivos do Unity e escolha o arquivo `.tgz` do Server SDK do Amazon GameLift. Escolha Abrir para instalar o plug-in.

O SDK do servidor do Amazon GameLift usa a estrutura log4net para gerar mensagens de log. Ele é configurado para enviar mensagens para o terminal de um servidor criado por padrão, mas o Unity exige configuração para adicionar suporte ao registro de arquivos. Você poderá adicionar esse suporte ao seu projeto importando a amostra fornecida dentro do pacote do Server SDK do Amazon GameLift. Use o procedimento a seguir para adicionar a amostra e configurar o log4net:

Para configurar o log4net para saída de arquivo

1. Com seu projeto aberto no Unity Editor, vá até o menu principal e selecione Window, Gerenciador de pacotes.
2. No menu suspenso, selecione Pacotes: No projeto e, em seguida, selecione Server SDK do Amazon GameLift na lista de pacotes. Isso abre os detalhes do pacote.
3. Nos detalhes do pacote, selecione a opção Grupo de amostras e pressione Importar.
4. O arquivo `log4net.config` e o script `LoggingConfiguration.cs` que o acompanha executam automaticamente a configuração, que agora está configurada na pasta do projeto `Assets/Samples`.

Note

Se precisar mover seu arquivo `log4net.config` para uma pasta diferente no projeto, você também deverá atualizar o caminho do arquivo de configuração no script `LoggingConfiguration.cs` com o novo caminho. Para obter mais informações, consulte o [manual do log4net sobre como configurar o log4net](#).

Para obter instruções mais detalhadas e orientações de teste, consulte o link README localizado no download do plug-in.

Configure uma frota Anywhere do Amazon GameLift para testes

Você poderá configurar sua estação de trabalho de desenvolvimento como uma frota de hospedagem do Anywhere do Amazon GameLift para testar iterativamente sua integração com o Amazon GameLift. Com essa configuração, você pode iniciar os processos do servidor de jogos em sua estação de trabalho, enviar solicitações de junção de jogadores ou de marcação de jogos ao Amazon GameLift para iniciar sessões de jogo e conectar clientes às novas sessões de jogo. Com sua própria estação de trabalho configurada como servidor de hospedagem, você pode monitorar todos os aspectos da integração do seu jogo com o Amazon GameLift.

Para obter instruções sobre como configurar sua estação de trabalho, consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#) para concluir as seguintes etapas:

1. Crie um local personalizado para sua estação de trabalho.
2. Crie uma frota Anywhere do Amazon GameLift com sua nova localização personalizada. Se for bem-sucedida, essa solicitação retornará uma ID da frota. Anote esse valor, pois você precisará disso mais tarde.
3. Registre sua estação de trabalho como uma computação na nova frota Anywhere. Forneça um nome de computação exclusivo e especifique o endereço IP da sua estação de trabalho. Se for bem-sucedida, essa solicitação retornará um endpoint do SDK de serviço, na forma de uma URL do WebSocket. Anote esse valor, pois você precisará disso mais tarde.
4. Gere um token de autenticação para a computação da sua estação de trabalho. Essa autenticação de curta duração inclui o token e uma data de expiração. Seu servidor de jogo o usa para autenticar a comunicação com o serviço do Amazon GameLift. Armazene a autenticação no computador da sua estação de trabalho para que os processos do servidor de jogos em execução possam acessá-la.

Adicione o código do servidor do Amazon GameLift ao seu projeto do Unity

Seu servidor de jogo se comunica com o serviço do Amazon GameLift para receber instruções e relatar o status contínuo. Para fazer isso, você adiciona um código de servidor de jogos que usa o SDK do servidor do Amazon GameLift.

O exemplo de código fornecido ilustra os elementos básicos de integração necessários. Ele usa um `MonoBehavior` para ilustrar uma inicialização simples do servidor de jogos com o Amazon GameLift. O exemplo pressupõe que o servidor do jogo seja executado em uma frota Anywhere do Amazon GameLift para testes. Inclui código para:

- Inicialize um cliente da API Amazon GameLift. A amostra usa a versão de `InitSDK()` com parâmetros do servidor para sua frota Anywhere e computação. Use o URL do WebSocket, o ID da frota, o nome do computador (ID do host) e o token de autenticação, conforme definido no tópico anterior [Configure uma frota Anywhere do Amazon GameLift para testes](#).
- Implemente funções de retorno de chamada para responder às solicitações do serviço do Amazon GameLift, inclusive `OnStartGameSession`, `OnProcessTerminate` e `onHealthCheck`.
- Chame `processReady()` com uma porta designada para notificar o serviço do Amazon GameLift quando o processo estiver pronto para hospedar sessões de jogo.

O código apresentado neste tópico estabelece comunicação com o serviço do Amazon GameLift e . Ele também implementa um conjunto de funções de retorno de chamada que respondem às solicitações do . Para obter mais informações sobre cada função e o que o código faz, consulte [Inicializar o processo do servidor](#). Para obter mais informações sobre as ações do SDK e os tipos de dados usados nesse código, leia [Referência do SDK do servidor do Amazon GameLift para C#](#).

Este exemplo mostra como adicionar todos os elementos necessários, conforme descrito em [Adicionar o Amazon GameLift ao seu servidor de jogos](#). Inclui:

Para obter mais informações sobre como adicionar a funcionalidade do Amazon GameLift, consulte estes tópicos:

- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Referência do SDK do servidor do Amazon GameLift para C#](#)

```
using System.Collections.Generic;
using Aws.GameLift.Server;
using UnityEngine;

public class ServerSDKManualTest : MonoBehaviour
{
    //This example is a simple integration that initializes a game server process
    //that is running on an Amazon GameLift Anywhere fleet.
    void Start()
    {
        //Identify port number (hard coded here for simplicity) the game server is
        //listening on for player connections
        var listeningPort = 7777;

        //WebSocketUrl from RegisterHost call
        var websocketUrl = "wss://us-west-2.api.amazongamelift.com";

        //Unique identifier for this process
        var processId = "myProcess";

        //Unique identifier for your host that this process belongs to
        var hostId = "myHost";

        //Unique identifier for your fleet that this host belongs to
        var fleetId = "myFleet";
    }
}
```

```
//Authorization token for this host process
var authToken = "myAuthToken";

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

//InitSDK establishes a local connection with an Amazon GameLift agent
//to enable further communication.
var initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);
if (initSDKOutcome.Success)
{
    //Implement callback functions
    ProcessParameters processParameters = new ProcessParameters(
        //Implement OnStartGameSession callback
        (gameSession) => {
            //GameLift sends a game session activation request to the game
server
            //with game session object containing game properties and other
settings.
            //Here is where a game server takes action based on the game
session object.
            //When the game server is ready to receive incoming player
connections,
            //it invokes the server SDK call ActivateGameSession().
            GameLiftServerAPI.ActivateGameSession();
        },
        (updateGameSession) => {
            //GameLift sends a request when a game session is updated (such as
for
            //FlexMatch backfill) with an updated game session object.
            //The game server can examine matchmakerData and handle new
incoming players.
            //updateReason explains the purpose of the update.
        },
        () => {
            //Implement callback function OnProcessTerminate
            //GameLift invokes this callback before shutting down the instance
hosting this game server.
        }
    );
}
```

```

        //It gives the game server a chance to save its state, communicate
with services, etc.,
        //and initiate shut down. When the game server is ready to shut
down, it invokes the
        //server SDK call ProcessEnding() to tell GameLift it is shutting
down.
        GameLiftServerAPI.ProcessEnding();
    },
    () => {
        //Implement callback function OnHealthCheck
        //GameLift invokes this callback approximately every 60 seconds.
        //A game server might want to check the health of dependencies,
etc.
        //Then it returns health status true if healthy, false otherwise.
        //The game server must respond within 60 seconds, or GameLift
records 'false'.
        //In this example, the game server always reports healthy.
        return true;
    },
    //The game server gets ready to report that it is ready to host game
sessions
    //and that it will listen on port 7777 for incoming player connections.
    listeningPort,
    new LogParameters(new List<string>()
    {
        //Here, the game server tells GameLift where to find game session
log files.
        //At the end of a game session, GameLift uploads everything in the
specified
        //location and stores it in the cloud for access later.
        "/local/game/logs/myserver.log"
    }));

    //The game server calls ProcessReady() to tell GameLift it's ready to host
game sessions.
    var processReadyOutcome =
GameLiftServerAPI.ProcessReady(processParameters);
    if (processReadyOutcome.Success)
    {
        print("ProcessReady success.");
    }
    else
    {

```

```
        print("ProcessReady failure : " +
processReadyOutcome.Error.ToString());
    }
}
else
{
    print("InitSDK failure : " + initSDKOutcome.Error.ToString());
}
}

void OnApplicationQuit()
{
    //Make sure to call GameLiftServerAPI.ProcessEnding() and
GameLiftServerAPI.Destroy() before terminating the server process.
    //These actions notify Amazon GameLift that the process is terminating and
frees the API client from memory.
    GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();
    GameLiftServerAPI.Destroy();
    if (processEndingOutcome.Success)
    {
        Environment.Exit(0);
    }
    else
    {
        Console.WriteLine("ProcessEnding() failed. Error: " +
processEndingOutcome.Error.ToString());
        Environment.Exit(-1);
    }
}
}
```

Recursos adicionais

Use os seguintes recursos para testar seu servidor de jogos e expandir a funcionalidade:

- Configure sua máquina de desenvolvimento como uma frota Anywhere do Amazon GameLift e use-a para testes locais. Consulte [Teste sua integração personalizada com o servidor](#).
- Crie seu servidor de jogo e faça o upload da compilação para o Amazon GameLift. Consulte [Carregar uma versão de servidor personalizada para o Amazon GameLift](#).
- Implante sua versão de servidor de jogo em uma frota EC2 gerenciada pelo Amazon GameLift. Consulte [Criar uma nova frota do Amazon GameLift](#).

Teste sua integração usando frotas Anywhere do Amazon GameLift

Você pode usar uma frota Anywhere do Amazon GameLift para criar e testar iterativamente sua integração de jogos com o Amazon GameLift. Configure seu próprio hardware como uma frota Anywhere com uma conexão com o serviço Amazon GameLift e, em seguida, instale e execute seu servidor de jogos nele. Use um aplicativo de teste para executar cenários como iniciar/interromper sessões de jogo, rastrear conexões de jogadores e processar alocações de marcação de jogos. Com uma frota Anywhere, você poderá atualizar a versão do servidor de jogos conforme necessário e ter visibilidade total da atividade de hospedagem.

Você poderá usar frotas Anywhere do Amazon GameLift com jogos integrados ao Server SDK do Amazon GameLift versão 5 ou superior.

Tópicos

- [Desenvolvimento inicial](#)
- [Faça iterações em seu servidor de jogos](#)

Desenvolvimento inicial

Você desenvolveu seu jogo e o está integrando ao SDK de servidor do Amazon GameLift. Para testar sua integração, você pode fazer o upload de cada nova iteração do seu servidor de jogos para o Amazon GameLift e criar uma frota. Como alternativa, usar uma frota Anywhere com seu laptop de desenvolvimento oferece uma maneira mais eficiente de realizar desenvolvimento e testes iterativos.

Use os procedimentos a seguir para criar uma frota Anywhere e iniciar uma sessão de jogo em seu laptop usando o console do Amazon GameLift ou AWS Command Line Interface (AWS CLI).

Console

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, em Hospedagem, escolha Localizações.
3. Selecione a opção Criar local.
4. Na caixa de diálogo Criar local, faça o seguinte:
 - a. Insira um Nome de localização. Isso rotula a localização dos seus recursos computacionais que o Amazon GameLift usa para executar seus jogos em frotas Anywhere. Os nomes de locais personalizados devem começar com custom-.
 - b. Selecione a opção Criar.

5. Para criar uma frota Anywhere, você poderá fazer o seguinte:
 - a. No painel de navegação, em Hospedagem, escolha Frotas.
 - b. Na página Frotas, selecione Criar frota.
 - c. Na etapa Escolher tipo de computação, escolha Anywhere e, em seguida, escolha Avançar.
 - d. Na etapa Definir detalhes da frota, defina sua nova frota. Para obter mais informações, consulte [Crie uma frota do Amazon GameLift](#).
 - e. Na etapa Selecionar locais, selecione o local personalizado que você criou.
 - f. Conclua as etapas restantes de criação da frota para criar sua frota Anywhere.
6. Registre seu laptop como um recurso de computação na frota que você criou. Use o comando [register-compute](#) (ou a operação [RegisterCompute](#) da API). Inclua a `fleet-id` criado na etapa anterior e adicione um `compute-name` e o `ip-address` do seu laptop.

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

Exemplos de resultado:

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

7. Inicie uma sessão de depuração do seu servidor de jogos.
 - a. Obtenha o token de autorização para seu laptop na frota que você criou. Use o comando [get-compute-auth-token](#) (ou a operação [GetComputeAuthToken](#) da API).

```
aws gamelift get-compute-auth-token \  
  --fleet-id fleet-1234 \  
  --location custom-location-1
```



```
--compute-name DevLaptop
```

Exemplos de resultado:

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

- b. Execute uma instância de depuração do executável do seu servidor de jogos. Para executar a instância de depuração, seu servidor de jogo deve chamar [InitSDK\(\)](#). Depois que o processo estiver pronto para hospedar uma sessão de jogo, o servidor do jogo chama [ProcessReady\(\)](#).
8. Crie uma sessão de jogo para testar sua primeira integração com o Anywhere do Amazon GameLift. Use o comando [create-game-session](#) (ou a operação [CreateGameSession](#) da API). Especifique a localização personalizada da frota.

```
aws gamelift create-game-session \  
  --fleet-id fleet-1234 \  
  --name DebugSession \  
  --maximum-player-session-count 2 \  
  --location custom-location-1
```

Exemplos de resultado:

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

O Amazon GameLift envia uma mensagem `onStartGameSession()` para seu processo de servidor registrado. A mensagem contém o objeto `GameSession` da etapa anterior com propriedades do jogo, dados das sessões do jogo, dados do marcador de jogos e mais sobre a sessão do jogo.

9. Adicione lógica ao seu servidor de jogo para que o processo do servidor responda à mensagem `onStartGameSession()` com `ActivateGameSession()`. A operação envia uma confirmação ao Amazon GameLift de que seu servidor recebeu e aceitou a mensagem de criação da sessão de jogo. Para obter mais informações, consulte, [Referência do SDK do servidor do Amazon GameLift](#).

Seu servidor de jogo agora está executando uma sessão de jogo para você testar e usar para iteração. Para saber como iterar no servidor do jogo, siga para a próxima seção.

AWS CLI

1. Crie um local personalizado usando o comando [create-location](#) (ou a operação [CreateLocation](#) da API). Um local personalizado identifica a localização do seu hardware que o Amazon GameLift usa para executar seus jogos em frotas Anywhere.

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

Exemplos de resultado:

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

2. Crie uma frota Anywhere com sua localização personalizada usando o comando [create-fleet](#) (ou a operação [CreateFleet](#) da API). O Amazon GameLift cria a frota em sua região de origem e os locais personalizados que você fornece.

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

Exemplos de resultado:

```
Fleet {  
  Name = LaptopFleet,  
  ComputeType = ANYWHERE,
```

```
FleetId = fleet-1234,  
Status = ACTIVE  
...  
}
```

3. Registre seu laptop como um recurso computacional na frota que você criou. Use o comando [register-compute](#) (ou a operação [RegisterCompute](#) da API). Inclua o `fleet-id` criado na etapa anterior e adicione um `compute-name` e o `ip-address` público do seu laptop.

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

Exemplos de resultado:

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

4. Inicie uma sessão de depuração do seu servidor de jogos.
 - a. Obtenha o token de autorização para seu laptop na frota que você criou. Use o comando [get-compute-auth-token](#) (ou a operação [GetComputeAuthToken](#) da API).

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

Exemplos de resultado:

```
ComputeAuthToken {  
  FleetId = fleet-1234,  
  ComputeName = DevLaptop,
```

```
AuthToken = abcdefg123,  
ExpirationTime = 1897492857.11  
}
```

- b. Execute uma instância de depuração do executável do seu servidor de jogos. Para executar a instância de depuração, seu servidor de jogo deve chamar `InitSDK()`. Depois que o processo estiver pronto para hospedar uma sessão de jogo, o servidor do jogo chama `ProcessReady()`.
5. Crie uma sessão de jogo para testar sua primeira integração com o Anywhere do Amazon GameLift. Use o comando [create-game-session](#) (ou a operação [CreateGameSession](#) a API).

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

Exemplos de resultado:

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

O Amazon GameLift envia uma mensagem `onStartGameSession()` para seu processo de servidor registrado. A mensagem contém o objeto `GameSession` da etapa anterior com propriedades do jogo, dados das sessões do jogo, dados do marcador de jogos e mais sobre a sessão do jogo.

6. Adicione lógica ao seu servidor de jogo para que o processo do servidor responda à mensagem `onStartGameSession()` com `ActivateGameSession()`. A operação envia uma confirmação ao Amazon GameLift de que seu servidor recebeu e aceitou a mensagem de criação da sessão de jogo. Para obter mais informações, consulte, [Referência do SDK do servidor do Amazon GameLift](#).

Seu servidor de jogo agora está executando uma sessão de jogo para você testar e usar para iteração. Para saber como iterar no servidor do jogo, siga para a próxima seção.

Faça iterações em seu servidor de jogos

Nesse caso de uso, considere um cenário em que você configurou e testou seu servidor de jogos e encontrou um bug. Com o Anywhere do Amazon GameLift, você pode iterar seu código e evitar a configuração pesada de usar um Amazon EC2 Fleet.

1. Limpe a `GameSession` existente, se possível. Se o servidor do jogo travar ou não chamar `ProcessEnding()`, o Amazon GameLift limpa as `GameSession` depois que o servidor do jogo parar de enviar verificações de saúde.
2. Faça as alterações no código do seu servidor de jogo, compile e prepare-se para o próximo teste.
3. Sua frota Anywhere anterior ainda está ativa e seu laptop ainda está registrado como um recurso computacional na frota. Para começar a testar novamente, crie uma nova instância de depuração.
 - a. Recupere o token de autorização do seu laptop na frota que você criou. Use o comando [get-compute-auth-token](#) (ou a operação [GetComputeAuthToken](#) da API).

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

Exemplos de resultado:

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

- b. Execute uma instância de depuração do executável do seu servidor de jogos. Para executar a instância de depuração, seu servidor de jogo deve chamar `InitSDK()`. Depois que o processo estiver pronto para hospedar uma sessão de jogo, o servidor do jogo chama `ProcessReady()`.

4. Sua frota agora tem um processo de servidor disponível. Crie sua sessão de jogo e realize seus próximos testes. Use o comando [create-game-session](#) (ou a operação [CreateGameSession](#) da API).

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

O Amazon GameLift envia uma mensagem `onStartGameSession()` para seu processo de servidor registrado. A mensagem contém o objeto `GameSession` da etapa anterior com propriedades do jogo, dados da sessão do jogo, dados do marcador de jogos e mais sobre a sessão do jogo.

5. Adicione lógica ao seu servidor de jogo para que o processo do servidor responda à mensagem `onStartGameSession()` com `ActivateGameSession()`. A operação envia uma confirmação ao Amazon GameLift de que seu servidor recebeu e aceitou a mensagem de criação da sessão de jogo. Para obter mais informações, consulte, [Referência do SDK do servidor do Amazon GameLift](#).

Depois de terminar de testar seu servidor de jogos, você pode continuar usando o Amazon GameLift para gerenciar sua frota e servidores de jogos. Para obter mais informações, consulte [Crie uma GameLift Anywhere frota da Amazon](#).

Teste sua integração usando o Amazon GameLift Local

Note

Use esse procedimento de teste se você estiver usando uma versão do SDK de servidor do Amazon GameLift que seja 4.x ou anterior. O pacote de SDK do seu servidor inclui uma versão compatível do Amazon GameLift Local. Se você estiver usando o SDK do servidor versão 5.x, consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#) para testes locais com uma frota Anywhere do Amazon GameLift.

Use o Amazon GameLift Local para executar uma versão limitada do serviço gerenciado do Amazon GameLift em um dispositivo local e testar a integração do jogo com ele. Esta ferramenta é útil ao fazer o desenvolvimento iterativo na integração do jogo. A alternativa, fazer upload de cada nova

compilação para o Amazon GameLift e configurar uma frota para hospedar seu jogo, poderá levar 30 minutos ou mais a cada vez.

Com o Amazon GameLift Local, é possível verificar o seguinte:

- O servidor de jogos está integrado corretamente ao Server SDK e se comunicando corretamente com o serviço do Amazon GameLift para iniciar novas sessões de jogos, aceitar novos jogadores e relatar integridade e status.
- O cliente do jogo está integrado corretamente ao SDK da AWS para o Amazon GameLift e é capaz de recuperar informações sobre sessões de jogos existentes, iniciar novas sessões de jogos, adicionar jogadores a jogos e se conectar à sessão do jogo.

O Amazon GameLift Local é uma ferramenta da linha de comando que inicia uma versão autocontida do serviço do Amazon GameLift. O Amazon GameLift Local também fornece um log de eventos em execução de inicialização do processo do servidor, verificações de integridade e chamadas à API, além de respostas. O Amazon GameLift Local reconhece um subconjunto das ações do SDK da AWS para o Amazon GameLift. Você pode fazer chamadas pela AWS CLI ou pelo cliente do jogo. Todas as ações de API são realizadas localmente, assim como acontece no serviço Web do Amazon GameLift.

Cada processo do servidor deve hospedar apenas uma única sessão de jogo. A sessão do jogo é o executável que você usa para se conectar ao Amazon GameLift Local. Quando a sessão do jogo estiver concluída, você deverá chamar `GameLiftServerSDK::ProcessEnding` e sair do processo. Ao testar localmente com o Amazon GameLift Local, você pode iniciar vários processos de servidor. Cada processo se conectará ao Amazon GameLift Local. Em seguida, você pode criar uma sessão de jogo para cada processo do servidor. Quando sua sessão de jogo terminar, o processo do servidor de jogos deverá sair. Em seguida, você deve iniciar manualmente outro processo do servidor.

O Amazon GameLift local oferece suporte às seguintes APIs:

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

Configurar o Amazon GameLift Local

O Amazon GameLift Local é fornecido como um arquivo `.jar` executável que acompanha o [Server SDK](#). Ele pode ser executado no Windows ou no Linux e usado com qualquer linguagem com suporte pelo Amazon GameLift.

Para executar o Local, você também deve ter instalado os itens a seguir.

- Uma compilação do Server SDK do Amazon GameLift, versão 3.1.5 a 4.x.
- Java 8

Testar um servidor de jogos

Se quiser testar apenas o servidor de jogos, é possível usar a AWS CLI para simular chamadas do cliente do jogo ao serviço do Amazon GameLift Local. Isso verifica se o servidor de jogos está funcionando conforme esperado com o seguinte:

- O servidor de jogos é executado corretamente e inicializa o Server SDK do Amazon GameLift.
- Como parte do processo de inicialização, o servidor de jogos notifica o Amazon GameLift de que o servidor está pronto para hospedar sessões de jogos.
- O servidor de jogos envia o status da integridade para o Amazon GameLift a cada minuto durante a execução.
- O servidor de jogos responde a solicitações para iniciar uma nova sessão de jogo.

1. Inicie o Amazon GameLift Local.

Abra uma janela de prompt de comando, navegue até o diretório que contém o arquivo `GameLiftLocal.jar` e o execute. Por padrão, o Local escuta solicitações de clientes de jogos na porta 8080. Para especificar um número de porta diferente, use o parâmetro `-p`, conforme mostrado no seguinte exemplo:

```
java -jar GameLiftLocal.jar -p 9080
```

Após a inicialização do Local, você vê logs indicando que dois servidores locais foram iniciados, um escutando o servidor de jogos e um escutando o cliente do jogo ou a AWS CLI. Os logs continuam informando a atividade nos dois servidores locais, inclusive a comunicação com e de componentes do jogo.

2. Inicie o servidor de jogos.

Inicie o servidor de jogos integrado ao Amazon GameLift localmente. Você não precisa alterar o endpoint do servidor de jogos.

Na janela do prompt de comando do Local, as mensagens de log indicam que o servidor de jogos se conectou ao serviço do Amazon GameLift Local. Isso significa que o servidor de jogos inicializou com êxito o Server SDK do Amazon GameLift (com `InitSDK()`). Ele chamou `ProcessReady()` com os caminhos de log mostrados e, se for bem-sucedido, estará pronto para hospedar uma sessão de jogo. Embora o servidor de jogos esteja em execução, o Amazon GameLift registra em log o relatório de status da integridade do servidor de jogos. O seguinte exemplo do sistema de mensagens de log mostra um servidor de jogos integrado com êxito:

```
16:50:53,217 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - SDK
connected: /127.0.0.1:64247
16:50:53,217 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - SDK pid is 17040,
sdkVersion is 3.1.5 and sdkLanguage is CSharp
16:50:53,217 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - NOTE: Only SDK
versions 3.1.5 and above are supported in GameLiftLocal!
16:50:53,451 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onProcessReady
received from: /127.0.0.1:64247 and ackRequest requested? true
16:50:53,543 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onProcessReady
data: logPathsToUpload: "C:\\game\\logs"
logPathsToUpload: "C:\\game\\error"
port: 1935

16:50:53,544 INFO || - [HostProcessManager] nioEventLoopGroup-3-1 - Registered new
process true, true,
16:50:53,558 INFO || - [SDKListenerImpl] nioEventLoopGroup-3-1 - onReportHealth
received from /127.0.0.1:64247 with health status: healthy
```

Entre as mensagens de erro e aviso em potencial estão as seguintes:

- Erro: "ProcessReady não encontrou um processo com pID: *<ID do processo>*! O `InitSDK()` foi invocado?"
- Aviso: "ProcessReady não encontrou um processo com pID: *<ID do processo>*! `ProcessReady(...)` é invocado mais de uma vez?"

3. Inicie a AWS CLI.

Assim que o servidor de jogos chamar `ProcessReady()` com êxito, você poderá começar a fazer chamadas de cliente. Abra outra janela do prompt de comando e inicie a ferramenta AWS CLI. A AWS CLI, por padrão, usa o endpoint de web service do Amazon GameLift. Você deve substituí-lo pelo endpoint do Local em todas as solicitações usando o parâmetro `--endpoint-url`, conforme mostrado na solicitação de exemplo a seguir.

```
AWS gamelift describe-game-sessions --endpoint-url http://localhost:9080 --fleet-id fleet-123
```

Na janela do prompt de comando da AWS CLI, os comandos `AWS gamelift` resultam em respostas conforme documentado em [Referência de comando da AWS CLI](#).

4. Crie uma sessão de jogo.

Com a AWS CLI, envie uma solicitação [CreateGameSession\(\)](#). A solicitação deve seguir a sintaxe esperada. Para Local, o parâmetro `FleetId` pode ser definido como qualquer string válida (`^fleet-\S+`).

```
AWS gamelift create-game-session --endpoint-url http://localhost:9080 --maximum-player-session-count 2 --fleet-id fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d
```

Na janela do prompt de comando do Local, as mensagens de log indicam que o Amazon GameLift Local enviou para o servidor de jogos uma chamada de retorno `onStartGameSession`. Se uma sessão de jogo for criada com êxito, o servidor de jogos responderá invocando `ActivateGameSession`.

```
13:57:36,129 INFO || - [SDKInvokerImpl]
    Thread-2 - Finished sending event to game server to start a game session:
    arn:aws:gamelift:local::gamesession/
    fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-ab423a4b-b827-4765-
    aea2-54b3fa0818b6.
    Waiting for ack response.13:57:36,143 INFO || - [SDKInvokerImpl]
    Thread-2 - Received ack response: true13:57:36,144 INFO || -
    [CreateGameSessionDispatcher] Thread-2 - GameSession with id:
    arn:aws:gamelift:local::gamesession/
    fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-ab423a4b-b827-4765-
    aea2-54b3fa0818b6
    created13:57:36,227 INFO || - [SDKListenerImpl]
```

```
nioEventLoopGroup-3-1 - onGameSessionActivate received
from: /127.0.0.1:60020 and ackRequest
requested? true13:57:36,230 INFO || - [SDKListenerImpl]
nioEventLoopGroup-3-1 - onGameSessionActivate data: gameId:
"arn:aws:gamelift:local::gamesession/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-abcdef12-3456-7890-abcd-
ef1234567890"
```

Na janela da AWS CLI, o Amazon GameLift responde com um objeto de sessão do jogo, incluindo um ID de sessão do jogo. O novo status da sessão do jogo é Ativação. O status mudará para Ativo assim que o servidor de jogos invocar `ActivateGameSession`. Se você quiser ver o status alterado, use a AWS CLI para chamar `DescribeGameSessions()`.

```
{
  "GameSession": {
    "Status": "ACTIVATING",
    "MaximumPlayerSessionCount": 2,
    "FleetId": "fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d",
    "GameSessionId": "arn:aws:gamelift:local::gamesession/
fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d/gsess-abcdef12-3456-7890-abcd-
ef1234567890",
    "IpAddress": "127.0.0.1",
    "Port": 1935
  }
}
```

Testar um servidor e um cliente de jogo

Para verificar a integração completa do jogo, inclusive a conexão de jogadores com jogos, você pode executar o servidor de jogos e o cliente localmente. Isso permite testar chamadas programáticas pelo cliente do jogo para o Amazon GameLift Local. Você pode verificar as seguintes ações:

- O cliente do jogo está fazendo solicitações do SDK da AWS com êxito para o serviço do Amazon GameLift Local, inclusive para criar sessões de jogos, recuperar informações sobre sessões de jogos existente e criar sessões de jogadores.
- O servidor de jogos está validando corretamente jogadores quando eles tentam participar de uma sessão do jogo. Para jogadores validados, o servidor de jogos pode recuperar dados do jogador (se implementado).
- O servidor de jogos informa uma conexão perdida quando um jogador deixa o jogo.

- O servidor de jogos informa o encerramento de uma sessão do jogo.

1. Inicie o Amazon GameLift Local.

Abra uma janela de prompt de comando, navegue até o diretório que contém o arquivo *GameLiftLocal.jar* e o execute. Por padrão, o Local escuta solicitações de clientes de jogos na porta 8080. Para especificar um número de porta diferente, use o parâmetro `-p`, conforme mostrado no seguinte exemplo.

```
./gamelift-local -p 9080
```

Após a inicialização do Local, você vê logs mostrando que dois servidores locais foram iniciados, um escutando o servidor de jogos e um escutando o cliente do jogo ou a AWS CLI.

2. Inicie o servidor de jogos.

Inicie o servidor de jogos integrado ao Amazon GameLift localmente. Consulte [Testar um servidor de jogos](#) para obter mais detalhes sobre logs de mensagens.

3. Configure o cliente do jogo do Local e o inicie.

Para usar o cliente do jogo com o serviço do Amazon GameLift Local, é necessário fazer as seguintes alterações na configuração do cliente do jogo, conforme descrito em [Configure a Amazon GameLift em um serviço de back-end](#):

- Altere o objeto `ClientConfiguration` a fim de apontar para o endpoint do Local, como `http://localhost:9080`.
- Defina um valor de ID da frota de destino. Para Local, você não precisa de um ID de frota real; defina a frota de destino como qualquer string válida (`^fleet-\S+`), como `fleet-1a2b3c4d-5e6f-7a8b-9c0d-1e2f3a4b5c6d`.
- Defina credenciais da AWS. Para Local, você não precisa de credenciais da AWS reais; você pode definir a chave de acesso e a chave secreta como qualquer string.

Na janela do prompt de comando do Local, assim que você iniciar o cliente do jogo, as mensagens de log deverão indicar que ele inicializou o `GameLiftClient` e que se comunicou com êxito com o serviço do Amazon GameLift.

4. Teste as chamadas do cliente de jogo para o serviço do Amazon GameLift.

Verifique se o cliente do jogo foi bem-sucedido fazendo qualquer uma ou todas as seguintes chamadas à API:

- [CreateGameSession\(\)](#)
- [DescribeGameSessions\(\)](#)
- [CreatePlayerSession\(\)](#)
- [CreatePlayerSessions\(\)](#)
- [DescribePlayerSessions\(\)](#)

Na janela do prompt de comando do Local, somente chamadas para `CreateGameSession()` resultam em mensagens de log. Mensagens de log mostram quando o Amazon GameLift Local solicita que o servidor de jogos inicie uma sessão do jogo (retorno de chamada `onStartGameSession`) e recebe um `ActivateGameSession` bem-sucedido quando o servidor de jogos o invoca. Na janela da AWS CLI, todas as chamadas à API resultam em respostas ou mensagens de erro conforme documentado.

5. Verifique se o servidor de jogos está validando novas conexões de jogador.

Depois de criar uma sessão de jogo e uma sessão de jogador, estabeleça uma conexão direta com a sessão do jogo.

Na janela do prompt de comando do Local, as mensagens de log devem mostrar que o servidor de jogos enviou uma solicitação `AcceptPlayerSession()` para validar a nova conexão do jogador. Se você usar a AWS CLI para chamar `DescribePlayerSessions()`, o status da sessão do jogador deverá mudar de Reservado para Ativo.

6. Verifique se o servidor de jogos está informando o status do jogo e do jogador para o serviço do Amazon GameLift.

Para que o Amazon GameLift gerencie a demanda do jogador e informe corretamente as métricas, o servidor de jogos deve informar diversos status ao Amazon GameLift. Verifique se o Local está registrando em log eventos relacionados às ações a seguir. Também convém usar a AWS CLI para rastrear alterações feitas no status.

- O jogador se desconecta de uma sessão de jogo – As mensagens de log do Amazon GameLift Local devem mostrar que o servidor de jogos chama `RemovePlayerSession()`. Uma chamada à AWS CLI para `DescribePlayerSessions()` deve refletir uma alteração feita no status de `Active` para `Completed`. Também convém chamar

`DescribeGameSessions()` para verificar se a contagem de jogadores atual da sessão do jogo diminuiu em um.

- A sessão do jogo termina – As mensagens de log do Amazon GameLift Local devem mostrar que o servidor de jogos chama `TerminateGameSession()`.

Note

A orientação anterior era chamar `TerminateGameSession()` ao encerrar uma sessão de jogo. Esse método foi descontinuado com o Server SDK do Amazon GameLift v4.0.1. Consulte [Encerrar uma sessão de jogo](#).

- O processo do servidor foi finalizado – As mensagens de log do Amazon GameLift Local devem mostrar que o servidor de jogos chama `ProcessEnding()`. Uma chamada à AWS CLI para `DescribeGameSessions()` deve refletir uma alteração feita no status de `Active` para `Terminated` (ou `Terminating`).

Variações com o local

Ao usar o Amazon GameLift Local, lembre-se do seguinte:

- Diferentemente do web service do Amazon GameLift, o Local não rastreia o status da integridade do servidor e inicia o retorno de chamada `onProcessTerminate`. O Local simplesmente deixa de registrar em log relatórios de integridade do servidor de jogos.
- Para chamadas para o AWS SDK, os IDs da frota não são validados e podem ser qualquer valor de string que atenda aos requisitos de parâmetro (`^fleet-\S+`).
- Os IDs da sessão do jogo criada com o Local têm uma estrutura diferente. Eles incluem a string `local`, conforme mostrado aqui:

```
arn:aws:gamelift:local::gamesession/fleet-123/gsess-56961f8e-  
db9c-4173-97e7-270b82f0daa6
```

Integração de jogos com Servidores em Tempo Real do Amazon GameLift

Este tópico fornece uma visão geral da solução gerenciada Amazon GameLift com servidores em tempo real. A visão geral explica quando essa solução é adequada ao seu jogo e como os servidores em tempo real oferecem suporte a jogos multijogador.

Para obter um roteiro completo para a execução de seu jogo, consulte [Roteiro de hospedagem gerenciada do Amazon GameLift](#).

Tip

Para experimentar a hospedagem do servidor de jogos Amazon GameLift, consulte [Conceitos básicos do Amazon GameLift](#).

O que são servidores em tempo real?

Servidores em tempo real são servidores de jogos leves e prontos que o Amazon GameLift fornece para uso com seus jogos multijogador. Os servidores em tempo real removem o processo de desenvolvimento, teste e implantação de um servidor de jogos personalizado. Essa solução pode ajudar a minimizar o tempo e o esforço necessários para concluir o jogo.

Principais atributos

- Pilha de rede completa para interação entre servidores e clientes de jogos
- Funcionalidade principal do servidor de jogos
- Lógica de servidor personalizável
- Atualizações ao vivo das configurações em tempo real e lógica de servidor
- Criação de jogos do FlexMatch
- Controle flexível de recursos de hospedagem

Configure os servidores criando uma frota e fornecendo um script de configuração. Para obter mais informações sobre a criação de servidores em tempo real e como preparar o cliente do jogo, consulte [Preparar seu servidor em tempo real](#).

Como os servidores em tempo real gerenciam as sessões de jogo

Você pode adicionar lógica personalizada para gerenciamento da sessão de jogo, incorporando-a no script em tempo real. Você pode gravar o código para acessar objetos específicos do servidor, adicionar lógica orientada por eventos usando retornos de chamada ou adicionar lógica com base em cenários não eventos.

Como servidores e clientes em tempo real interagem

Durante uma sessão de jogo, os clientes de jogos interagem enviando mensagens para o servidor em tempo real por meio de um serviço de back-end. Em seguida, o serviço de back-end retransmite as mensagens entre os clientes de jogos para trocar atividade, estado do jogo e dados relevantes do jogo.

Além disso, você pode personalizar como os clientes e servidores interagem adicionando lógica de jogo ao script em tempo real. Com a lógica de jogo personalizada, um servidor em tempo real pode implementar retornos de chamada para acionar respostas orientadas por eventos.

Protocolo de comunicação

Os servidores em tempo real e os clientes de jogos conectados se comunicam por meio de dois canais: uma conexão TCP para entrega confiável e um canal UDP para entrega rápida. Ao criar mensagens, os clientes de jogos escolhem qual protocolo usar de acordo com a natureza da mensagem. A entrega da mensagem é definida como UDP por padrão. Se um canal UDP não estiver disponível, o Amazon GameLift enviará mensagens usando TCP como alternativa.

Conteúdo da mensagem

O conteúdo da mensagem consiste em dois elementos: um código de operação necessário (opCode) e uma carga opcional. O opCode de uma mensagem identifica um determinado evento de jogo ou atividade do jogador e a carga fornece dados adicionais relacionados ao código da operação. Ambos os elementos são definidos pelo desenvolvedor. O cliente do jogo age com base nos opCodes das mensagens que recebe.

Grupos de jogadores

Os servidores em tempo real fornecem funcionalidade para gerenciamento de grupos de jogadores. Por padrão, o Amazon GameLift posiciona todos os jogadores que estiverem conectados a um jogo em um grupo "todos os jogadores". Além disso, os desenvolvedores podem configurar outros

grupos para seus jogos, e os jogadores podem ser membros de vários grupos simultaneamente. Os membros do grupo podem enviar mensagens e compartilhar dados de jogos com todos os jogadores do grupo. Um uso possível para grupos é configurar equipes de jogadores e gerenciar a comunicação da equipe.

Servidores em tempo real com certificados TLS

Com os servidores em tempo real, a autenticação do servidor e a criptografia de pacotes de dados são incorporadas ao serviço. Esses atributos de segurança são ativados quando você ativa a geração do certificado TLS. Quando o cliente do jogo tenta se conectar a um servidor em tempo real, o servidor responde automaticamente com o certificado TLS, que o cliente valida. O Amazon GameLift processa a criptografia usando TLS para comunicação TCP (Websockets) e DTLS para tráfego UDP.

Personalizar um servidor em tempo real

Um servidor em tempo real funciona como um servidor de retransmissão sem estado. O servidor em tempo real retransmite pacotes de dados de jogos e mensagens entre os clientes de jogos conectados ao jogo. No entanto, o servidor em tempo real não avalia mensagens, processa dados nem executa qualquer lógica de jogo. Usado dessa forma, cada cliente do jogo mantém sua própria visualização do estado do jogo e fornece atualizações para outros jogadores por meio do servidor de retransmissão. Cada cliente é responsável por incorporar essas atualizações e conciliar seu próprio estado do jogo.

É possível personalizar os servidores adicionando à funcionalidade de script em tempo real. Com a lógica do jogo, por exemplo, é possível optar por criar um jogo com estado com uma visualização autoritativa do servidor do estado do jogo.

O Amazon GameLift define um conjunto de retornos de chamada do lado do servidor para scripts em tempo real. Implemente esses retornos de chamada para adicionar funcionalidade baseada em eventos a seu servidor. Por exemplo, é possível:

- Autenticar um jogador quando um cliente do jogo tentar se conectar ao servidor.
- Validar se um jogador pode ingressar em um grupo mediante solicitação.
- Determinar quando entregar mensagens de um determinado jogador ou para um jogador de destino ou executar processamento adicional em resposta.
- Notificar todos os jogadores quando um jogador deixa um grupo ou se desconecta do servidor.
- Visualizar o conteúdo dos objetos da sessão de jogo ou objetos de mensagem, e usar os dados.

Implantação e atualização de servidores em tempo real

Uma das principais vantagens dos servidores em tempo real é a capacidade de atualizar os scripts a qualquer momento. Ao atualizar um script, o Amazon GameLift distribui a nova versão para todos os recursos de hospedagem em alguns minutos. Depois que o Amazon GameLift implanta o novo script, todas as novas sessões de jogos criadas após esse ponto usarão a nova versão do script. (As sessões de jogo existentes continuarão usando a versão original.)

Comece a integrar seu jogo com os servidores em tempo real:

- [Integrar um cliente do jogo para servidores em tempo real](#)
- [Criar um script em tempo real](#)

Integrar um cliente do jogo para servidores em tempo real

Este tópico descreve como preparar o cliente do jogo para que ele possa ingressar e participar de sessões de jogo hospedadas pelo Amazon GameLift.

Há dois conjuntos de tarefas necessárias para preparar o cliente do jogo:

- Configure o cliente do jogo para que ele adquira informações sobre jogos existentes, solicite criação de jogos, inicie novas sessões de jogo e reserve slots de sessões de jogo para um jogador.
- Permita que o cliente do jogo participe de uma sessão de jogo hospedada em um servidor em tempo real e troque mensagens.

Encontrar ou criar sessões de jogo e sessões de jogadores

Configure o cliente do jogo para que ele descubra ou inicie sessões de jogo, solicite criação de jogos do FlexMatch e reserve espaço para jogadores em um jogo com a criação de sessões de jogador. Como melhor prática, crie um serviço de back-end e use-o para fazer solicitações diretas ao serviço do Amazon GameLift quando acionado por uma ação de cliente do jogo. O serviço de back-end transmite as respostas relevantes de volta para o cliente do jogo.

1. Adicione o SDK da AWS ao cliente do jogo, inicialize um cliente do Amazon GameLift e configure-o para que ele use os recursos de hospedagem em suas frotas e filas. O SDK da AWS está disponível em vários idiomas; consulte [Para serviços personalizados ao cliente](#) dos SDKs do Amazon GameLift.

2. Adicione a funcionalidade do GameLift ao serviço de back-end. Para instruções mais detalhadas, consulte [Adicione GameLift a Amazon ao seu cliente de jogos](#) [Adicionar a criação de jogos do FlexMatch](#). A melhor prática é o uso dos posicionamentos de sessões de jogo para criar novas sessões de jogo. Esse método permite que você aproveite a capacidade do GameLift para posicionar novas sessões de jogo de forma rápida e inteligente, bem como para usar dados de latência do jogador para minimizar o atraso de jogo. No mínimo, o serviço de back-end deve poder solicitar novas sessões de jogo e processar dados da sessão de jogo em resposta. Você também pode adicionar funcionalidade para pesquisar e obter informações sobre sessões de jogo existentes e solicitar sessões de jogadores, o que efetivamente reserva um slot de jogador em uma sessão de jogo existente.
3. Transmita informações sobre a conexão de volta para o cliente do jogo. O serviço de back-end recebe objetos de sessões de jogo e sessões de jogadores em resposta a solicitações para o serviço do Amazon GameLift. Esses objetos contêm informações, especialmente os detalhes da conexão (endereço IP e porta) e o ID da sessão de jogadores, que o cliente do jogo precisa para se conectar à sessão de jogo em execução em um servidor em tempo real.

Conecte-se a jogos em servidores em tempo real

Permita que o cliente do jogo se conecte diretamente a uma sessão de jogo hospedada em um servidor em tempo real e troque mensagens com o servidor e com outros jogadores.

1. Obtenha o SDK do cliente em tempo real, compile-o e adicione-o ao projeto do cliente do jogo. Consulte o arquivo README para obter mais informações sobre requisitos de SDK e instruções sobre como criar as bibliotecas de cliente.
2. Chame [Client\(\)](#) com uma configuração de cliente que especifica o tipo de conexão cliente/servidor a ser usada.

Note

Se estiver se conectando a um servidor em tempo real em execução em uma frota protegida por um certificado TLS, você deverá especificar um tipo de conexão segura.

3. Adicione a seguinte funcionalidade ao cliente do jogo. Consulte [Referência de API \(C#\) do cliente dos servidores em tempo real](#) para obter mais informações.
 - Conectar-se e desconectar-se de um jogo
 - [Connect\(\)](#)

- [Disconnect\(\)](#)
 - Enviar mensagens a destinatários de destino
 - [SendMessage\(\)](#)
 - Receber e processar mensagens
 - [OnDataReceived\(\)](#)
 - Ingressar e sair de grupos de jogadores
 - [JoinGroup\(\)](#)
 - [RequestGroupMembership\(\)](#)
 - [LeaveGroup\(\)](#)
4. Configure manipuladores de eventos para retornos de chamada do cliente conforme necessário. Consulte [Referência da API \(C#\) do cliente dos servidores em tempo real: retornos de chamada assíncronos](#).

Ao trabalhar com frotas em tempo real com a geração de certificados TLS habilitada, o servidor é autenticado automaticamente usando o certificado TLS. O tráfego TCP e UDP é criptografado em trânsito para fornecer segurança na camada de transporte. O tráfego TCP é criptografado usando TLS 1.2 e o tráfego UDP é criptografado usando DTLS 1.2.

Exemplos de cliente do jogo

Cliente em tempo real básico (C#)

Este exemplo ilustra uma integração básica do cliente do jogo com o SDK do cliente em tempo real (C#). Como mostrado, o exemplo inicializa um objeto de cliente em tempo real, configura manipuladores de eventos e implementa o retorno de chamada do lado do cliente, se conecta a um servidor em tempo real, envia uma mensagem e se desconecta.

```
using System;
using System.Text;
using Aws.GameLift.Realtime;
using Aws.GameLift.Realtime.Event;
using Aws.GameLift.Realtime.Types;

namespace Example
{
    /**
     * An example client that wraps the GameLift Realtime client SDK
     */
}
```

```
*
* You can redirect logging from the SDK by setting up the LogHandler as such:
* ClientLogger.LogHandler = (x) => Console.WriteLine(x);
*
*/
class RealTimeClient
{
    public Aws.GameLift.Realtime.Client Client { get; private set; }

    // An opcode defined by client and your server script that represents a custom
message type
    private const int MY_TEST_OP_CODE = 10;

    /// Initialize a client for GameLift Realtime and connect to a player session.
    /// <param name="endpoint">The DNS name that is assigned to Realtime server</
param>
    /// <param name="remoteTcpPort">A TCP port for the Realtime server</param>
    /// <param name="listeningUdpPort">A local port for listening to UDP traffic</
param>
    /// <param name="connectionType">Type of connection to establish between client
and the Realtime server</param>
    /// <param name="playerSessionId">The player session ID that is assigned to the
game client for a game session </param>
    /// <param name="connectionPayload">Developer-defined data to be used during
client connection, such as for player authentication</param>
    public RealTimeClient(string endpoint, int remoteTcpPort, int listeningUdpPort,
ConnectionType connectionType,
        string playerSessionId, byte[] connectionPayload)
    {
        // Create a client configuration to specify a secure or unsecure connection
type
        // Best practice is to set up a secure connection using the connection type
RT_OVER_WSS_DTLS_TLS12.
        ClientConfiguration clientConfiguration = new ClientConfiguration()
        {
            // C# notation to set the field ConnectionType in the new instance of
ClientConfiguration
            ConnectionType = connectionType
        };

        // Create a Realtime client with the client configuration
        Client = new Client(clientConfiguration);

        // Initialize event handlers for the Realtime client
    }
}
```

```
Client.ConnectionOpen += OnOpenEvent;
Client.ConnectionClose += OnCloseEvent;
Client.GroupMembershipUpdated += OnGroupMembershipUpdate;
Client.DataReceived += OnDataReceived;

// Create a connection token to authenticate the client with the Realtime
server
// Player session IDs can be retrieved using AWS SDK for GameLift
ConnectionToken connectionToken = new ConnectionToken(playerSessionId,
connectionPayload);

// Initiate a connection with the Realtime server with the given connection
information
Client.Connect(endpoint, remoteTcpPort, listeningUdpPort, connectionToken);
}

public void Disconnect()
{
    if (Client.Connected)
    {
        Client.Disconnect();
    }
}

public bool IsConnected()
{
    return Client.Connected;
}

/// <summary>
/// Example of sending to a custom message to the server.
///
/// Server could be replaced by known peer Id etc.
/// </summary>
/// <param name="intent">Choice of delivery intent i.e. Reliable, Fast etc. </
param>
/// <param name="payload">Custom payload to send with message</param>
public void SendMessage(DeliveryIntent intent, string payload)
{
    Client.SendMessage(Client.NewMessage(MY_TEST_OP_CODE)
        .WithDeliveryIntent(intent)
        .WithTargetPlayer(Constants.PLAYER_ID_SERVER)
        .WithPayload(StringToBytes(payload)));
}
}
```

```
/**
 * Handle connection open events
 */
public void OnOpenEvent(object sender, EventArgs e)
{
}

/**
 * Handle connection close events
 */
public void OnCloseEvent(object sender, EventArgs e)
{
}

/**
 * Handle Group membership update events
 */
public void OnGroupMembershipUpdate(object sender, GroupMembershipEventArgs e)
{
}

/**
 * Handle data received from the Realtime server
 */
public virtual void OnDataReceived(object sender, DataReceivedEventArgs e)
{
    switch (e.OpCode)
    {
        // handle message based on OpCode
        default:
            break;
    }
}

/**
 * Helper method to simplify task of sending/receiving payloads.
 */
public static byte[] StringToBytes(string str)
{
    return Encoding.UTF8.GetBytes(str);
}

/**
```

```
    * Helper method to simplify task of sending/receiving payloads.
    */
    public static string BytesToString(byte[] bytes)
    {
        return Encoding.UTF8.GetString(bytes);
    }
}
```

Criar um script em tempo real

Para usar os servidores em tempo real para o jogo, é preciso fornecer um script (na forma de algum código JavaScript) para configurar e, opcionalmente, personalizar uma frota de servidores em tempo real. Este tópico abrange as principais etapas da criação de um script em tempo real. Assim que o script estiver pronto, faça upload dele no serviço Amazon GameLift e use-o para criar uma frota (consulte [Carregar um script de servidores em tempo real no Amazon GameLift](#)).

Para preparar um script para uso com os servidores em tempo real, adicione a seguinte funcionalidade ao script em tempo real.

Gerenciar o ciclo de vida da sessão de jogos (obrigatório)

No mínimo, um script em tempo real deve incluir a função `Init()`, que prepara o servidor em tempo real para iniciar uma sessão de jogo. Também é altamente recomendável que você também forneça uma maneira de encerrar sessões de jogos, para garantir que novas sessões de jogos possam continuar a ser iniciadas na frota.

A função de retorno de chamada `Init()`, quando chamada, recebe um objeto de sessão em tempo real, que contém uma interface para o servidor em tempo real. Consulte [Interface de servidores em tempo real](#) para obter mais detalhes sobre essa interface.

Para encerrar corretamente a uma sessão de jogo, o script também deve chamar a função `session.processEnding` em tempo real do servidor. Isso requer um mecanismo para determinar quando encerrar uma sessão. O código do script de exemplo ilustra um mecanismo simples que verifica se há conexões de jogador e aciona o encerramento da sessão de jogo quando nenhum jogador tiver se conectado à sessão por um período especificado.

Os servidores em tempo real com a configuração mais básica – inicialização e encerramento do processo de servidor – atua, essencialmente, como servidores de retransmissão sem estado. O servidor em tempo real retransmite mensagens e dados de jogos entre os clientes de jogos que estão conectados ao jogo, mas não realiza ação independente para processar dados ou executar

lógica. Opcionalmente, é possível adicionar a lógica do jogo, acionada por eventos de jogos ou outros mecanismos, conforme necessário para o jogo.

Adicionar lógica do jogo do lado do servidor (opcional)

Opcionalmente, é possível adicionar lógica do jogo ao script em tempo real. Por exemplo, você pode fazer qualquer um ou todos os seguintes. O código de script de exemplo fornece ilustração. Consulte [Referência de script de Servidores em Tempo Real do Amazon GameLift](#).

- Adicionar lógica orientada por eventos. Implemente as funções de retorno de chamada para responder a eventos de cliente-servidor. Consulte [Retornos de chamada de script para servidores em tempo real](#) para obter uma lista completa de retornos de chamada.
- Acione lógica enviando mensagens ao servidor. Crie um conjunto de códigos de operação especiais para mensagens enviadas de clientes de jogo para o servidor, e adicione funções para manipular o recebimento. Use o retorno de chamada `onMessage` e analise o conteúdo da mensagem usando a interface `gameMessage` (consulte [gameMessage.opcode](#)).
- Ative a lógica do jogo para acessar os outros recursos da AWS. Para obter mais detalhes, consulte [Comunique-se com outros recursos AWS de suas frotas](#).
- Permita que a lógica do jogo acesse as informações da frota da instância em que está sendo executada. Para obter mais detalhes, consulte [Obtenha dados da frota para uma instância do Amazon GameLift](#).

Exemplo de script de servidores em tempo real

Este exemplo ilustra um script básico necessário para implantar o s servidores em tempo real mais alguma lógica personalizada. Ele contém a função `Init()` necessária, e usa um mecanismo de temporizador para acionar o encerramento da sessão de jogo com base no tempo sem conexões de jogador. Ele também inclui alguns ganchos de lógica personalizada, incluindo algumas implementações de chamada de retorno.

```
// Example Realtime Server Script
'use strict';

// Example override configuration
const configuration = {
  pingIntervalTime: 30000,
  maxPlayers: 32
};
```

```
// Timing mechanism used to trigger end of game session. Defines how long, in
// milliseconds, between each tick in the example tick loop
const tickTime = 1000;

// Defines how long to wait in Seconds before beginning early termination check in
// the example tick loop
const minimumElapsedTime = 120;

var session; // The Realtime server session object
var logger; // Log at appropriate level
  via .info(), .warn(), .error(), .debug()
var startTime; // Records the time the process started
var activePlayers = 0; // Records the number of connected players
var onProcessStartedCalled = false; // Record if onProcessStarted has been called

// Example custom op codes for user-defined messages
// Any positive op code number can be defined here. These should match your client
// code.
const OP_CODE_CUSTOM_OP1 = 111;
const OP_CODE_CUSTOM_OP1_REPLY = 112;
const OP_CODE_PLAYER_ACCEPTED = 113;
const OP_CODE_DISCONNECT_NOTIFICATION = 114;

// Example groups for user-defined groups
// Any positive group number can be defined here. These should match your client code.
// When referring to user-defined groups, "-1" represents all groups, "0" is reserved.
const RED_TEAM_GROUP = 1;
const BLUE_TEAM_GROUP = 2;

// Called when game server is initialized, passed server's object of current session
function init(rtSession) {
  session = rtSession;
  logger = session.getLogger();
}

// On Process Started is called when the process has begun and we need to perform any
// bootstrapping. This is where the developer should insert any code to prepare
// the process to be able to host a game session, for example load some settings or set
// state
//
// Return true if the process has been appropriately prepared and it is okay to invoke
// the
// GameLift ProcessReady() call.
function onProcessStarted(args) {
```

```
    onProcessStartedCalled = true;
    logger.info("Starting process with args: " + args);
    logger.info("Ready to host games...");

    return true;
}

// Called when a new game session is started on the process
function onStartGameSession(gameSession) {
    // Complete any game session set-up

    // Set up an example tick loop to perform server initiated actions
    startTime = getTimeInS();
    tickLoop();
}

// Handle process termination if the process is being terminated by GameLift
// You do not need to call ProcessEnding here
function onProcessTerminate() {
    // Perform any clean up
}

// Return true if the process is healthy
function onHealthCheck() {
    return true;
}

// On Player Connect is called when a player has passed initial validation
// Return true if player should connect, false to reject
function onPlayerConnect(connectMsg) {
    // Perform any validation needed for connectMsg.payload, connectMsg.peerId
    return true;
}

// Called when a Player is accepted into the game
function onPlayerAccepted(player) {
    // This player was accepted -- let's send them a message
    const msg = session.newTextGameMessage(OP_CODE_PLAYER_ACCEPTED, player.peerId,
                                           "Peer " + player.peerId + " accepted");
    session.sendReliableMessage(msg, player.peerId);
    activePlayers++;
}

// On Player Disconnect is called when a player has left or been forcibly terminated
```

```
// Is only called for players that actually connected to the server and not those
// rejected by validation
// This is called before the player is removed from the player list
function onPlayerDisconnect(peerId) {
    // send a message to each remaining player letting them know about the disconnect
    const outMessage = session.newTextGameMessage(OP_CODE_DISCONNECT_NOTIFICATION,
                                                    session.getServerId(),
                                                    "Peer " + peerId + " disconnected");
    session.getPlayers().forEach((player, playerId) => {
        if (playerId !== peerId) {
            session.sendReliableMessage(outMessage, playerId);
        }
    });
    activePlayers--;
}

// Handle a message to the server
function onMessage(gameMessage) {
    switch (gameMessage.opCode) {
        case OP_CODE_CUSTOM_OP1: {
            // do operation 1 with gameMessage.payload for example sendToGroup
            const outMessage = session.newTextGameMessage(OP_CODE_CUSTOM_OP1_REPLY,
                session.getServerId(), gameMessage.payload);
            session.sendGroupMessage(outMessage, RED_TEAM_GROUP);
            break;
        }
    }
}

// Return true if the send should be allowed
function onSendToPlayer(gameMessage) {
    // This example rejects any payloads containing "Reject"
    return (!gameMessage.getPayloadAsText().includes("Reject"));
}

// Return true if the send to group should be allowed
// Use gameMessage.getPayloadAsText() to get the message contents
function onSendToGroup(gameMessage) {
    return true;
}

// Return true if the player is allowed to join the group
function onPlayerJoinGroup(groupId, peerId) {
    return true;
}
```

```
}

// Return true if the player is allowed to leave the group
function onPlayerLeaveGroup(groupId, peerId) {
    return true;
}

// A simple tick loop example
// Checks to see if a minimum amount of time has passed before seeing if the game has
// ended
async function tickLoop() {
    const elapsedTime = getTimeInS() - startTime;
    logger.info("Tick... " + elapsedTime + " activePlayers: " + activePlayers);

    // In Tick loop - see if all players have left early after a minimum period of time
    // has passed
    // Call processEnding() to terminate the process and quit
    if ( (activePlayers == 0) && (elapsedTime > minimumElapsedTime)) {
        logger.info("All players disconnected. Ending game");
        const outcome = await session.processEnding();
        logger.info("Completed process ending with: " + outcome);
        process.exit(0);
    }
    else {
        setTimeout(tickLoop, tickTime);
    }
}

// Calculates the current time in seconds
function getTimeInS() {
    return Math.round(new Date().getTime()/1000);
}

exports.ssExports = {
    configuration: configuration,
    init: init,
    onProcessStarted: onProcessStarted,
    onMessage: onMessage,
    onPlayerConnect: onPlayerConnect,
    onPlayerAccepted: onPlayerAccepted,
    onPlayerDisconnect: onPlayerDisconnect,
    onSendToPlayer: onSendToPlayer,
    onSendToGroup: onSendToGroup,
    onPlayerJoinGroup: onPlayerJoinGroup,
```

```
onPlayerLeaveGroup: onPlayerLeaveGroup,  
onStartGameSession: onStartGameSession,  
onProcessTerminate: onProcessTerminate,  
onHealthCheck: onHealthCheck  
};
```

Integrando jogos com o GameLift plugin Amazon para Unity

Os tópicos desta seção descrevem o GameLift plug-in da Amazon para Unity e como usá-lo para preparar seu projeto de jogo multijogador para hospedagem na Amazon GameLift. Trabalhe inteiramente em seu ambiente de desenvolvimento Unity com os fluxos de trabalho guiados do plug-in para completar os requisitos básicos de hospedagem na Amazon GameLift.

GameLift A Amazon é um serviço totalmente gerenciado que permite aos desenvolvedores de jogos gerenciar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões. Para obter mais informações sobre GameLift hospedagem na Amazon, consulte [Como o Amazon GameLift funciona](#).

- [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#), versão 2.0.0, funciona com o SDK 5.x do servidor e é compatível com a Amazon. GameLift Anywhere
- [Guia do GameLift plugin Amazon para Unity para servidor SDK 4.x](#), versão 1.0.0, funciona com o SDK 4.x do servidor ou anterior. Esta versão usa o Amazon GameLift Local para testes de integração.

Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x

GameLift A Amazon fornece ferramentas para preparar seus servidores de jogos multijogador para trabalhar com a Amazon GameLift. O GameLift plug-in da Amazon para Unity facilita a integração da Amazon GameLift em seus projetos de jogos Unity, o teste de sua integração com a Amazon GameLift Anywhere e a implantação de GameLift recursos da Amazon para hospedagem na nuvem.

Esse plug-in usa AWS CloudFormation modelos para implantar soluções de hospedagem para cenários comuns de jogos. Use essas soluções conforme fornecidas ou personalize-as conforme necessário para seus jogos.

Tópicos

- [Sobre o plug-in](#)

- [Fluxo do plug-in](#)
- [Instale o plug-in para Unity](#)
- [Configurar um perfil de usuário da AWS](#)
- [Configure seu jogo para testes locais com a Amazon GameLift Anywhere](#)
- [Implante seu jogo em hospedagem em nuvem com frotas de EC2 gerenciadas](#)

Sobre o plug-in

O plug-in para Unity fornece uma experiência inicial simplificada para integrar e hospedar seus jogos multijogador Unity com a Amazon. GameLift Você pode aproveitar a funcionalidade do plug-in e os componentes pré-criados para colocar seus jogos em funcionamento rapidamente.

O plug-in adiciona ferramentas e funcionalidades ao editor Unity. Use os fluxos de trabalho guiados para integrar a Amazon ao seu projeto de jogo, testá-lo localmente e, GameLift em seguida, implantar o servidor do jogo GameLift na hospedagem na nuvem da Amazon.

Use as soluções de hospedagem pré-criadas do plug-in para implantar seu jogo. Configure uma frota do Amazon GameLift Anywhere com sua estação de trabalho local como anfitriã. Para hospedagem na nuvem, escolha entre dois cenários de implantação comuns que equilibram a latência do jogador, a disponibilidade da sessão de jogo e o custo de maneiras diferentes. Um cenário inclui um simples FlexMatch matchmaker e um conjunto de regras. Use esses cenários para implementar uma solução básica de hospedagem pronta para produção e, em seguida, otimize e personalize conforme necessário.

O plug-in inclui os seguintes componentes:

- Módulos de plug-in para o editor Unity. Quando o plug-in é instalado, um novo item do menu principal dá acesso à GameLift funcionalidade da Amazon.
- Bibliotecas C# para a API de GameLift serviços da Amazon com funcionalidade do lado do cliente.
- Bibliotecas C# para o SDK GameLift do servidor Amazon (versão 5.x).
- Experimente o conteúdo do jogo, incluindo recursos e cenas, para que você possa experimentar a Amazon GameLift mesmo que não tenha um jogo multijogador pronto para ser construído.
- Configurações da solução, fornecidas como AWS CloudFormation modelos, que o plug-in usa ao implantar seu servidor de jogos na nuvem para hospedagem.

Fluxo do plug-in

As etapas a seguir descrevem uma abordagem típica para integrar e implantar um projeto de jogo com o GameLift plug-in da Amazon para Unity. Você conclui essas etapas trabalhando no editor Unity e no código do seu jogo.

1. Crie um perfil de usuário vinculado à sua AWS conta e forneça credenciais de acesso para um usuário de conta válido com permissões para usar a Amazon GameLift.
2. Adicione o código do servidor ao seu projeto de jogo para estabelecer comunicação entre um servidor de jogo em execução e o GameLift serviço com a Amazon.
3. Adicione o código do cliente ao seu projeto de jogo para permitir que os clientes do jogo enviem solicitações GameLift à Amazon para iniciar ou participar de uma sessão de jogo e depois se conectar ao servidor do jogo.
4. Use o fluxo de trabalho Anywhere para configurar sua estação de trabalho local como host Anywhere para seu servidor de jogos. Inicie seu servidor e cliente de jogos localmente, conecte-se a uma sessão de jogo e teste sua integração.
5. Use o fluxo de trabalho de hospedagem EC2 para carregar seu servidor de jogos integrado e implantar uma solução de hospedagem em nuvem. Quando seu servidor de jogo estiver pronto, inicie seu cliente de jogo localmente, conecte-se a uma sessão de jogo, faça login e jogue o jogo.

Ao trabalhar no plug-in, você criará e usará recursos AWS. Essas ações podem incorrer em cobranças na conta da AWS em uso. Se você é novatoAWS, as ações podem estar cobertas pelo [nível AWS gratuito](#).

Instale o plug-in para Unity

Esta seção descreve como adicionar o plug-in a um projeto Unity. Depois que o plug-in for instalado, a funcionalidade do plug-in estará disponível quando você tiver o projeto aberto no editor Unity.

Antes de começar

Veja o que você precisa para usar o GameLift plug-in da Amazon para Unity:

- Unity para Windows 2022 LTS ou Unity para macOS
- GameLift Plugin da Amazon para download do Unity. [\[Site de download\]](#) O download inclui dois pacotes:
 - Plugin GameLift autônomo da Amazon para Unity

- SDK do servidor Amazon GameLift C# para Unity
- Microsoft Visual Studio 2019 ou mais recente.
- Um projeto de jogo multijogador com código de jogo C#.
- O registro UnityNuGet com escopo de terceiros. Essa ferramenta gerencia DLLs de terceiros. Para obter mais informações, consulte o repositório [UnityNuGetGithub](#).

Adicione o plug-in ao projeto de jogo

Conclua as tarefas a seguir, trabalhando no editor Unity e nos arquivos de projeto do seu jogo.

Etapa 1: adicionar UnityNuGet ao seu projeto de jogo

Se você não tiver UnityNuGet configurado seu projeto de jogo, use as etapas a seguir para instalar a ferramenta usando o gerenciador de pacotes Unity. Como alternativa, você pode usar a NuGet CLI para baixar manualmente as DLLs. Para obter mais informações, consulte o Amazon GameLift C# server SDK for Unity. README

1. Com seu projeto aberto no editor Unity, acesse o menu principal e selecione Editar, Configurações do projeto. Nas opções, escolha a seção Gerenciador de pacotes e abra o grupo Registros com Escopo.
2. Escolha o botão + e insira os seguintes valores para o registro UnityNuGet com escopo:

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

Para usuários da versão 2021 do Unity:

Após a configuração UnityNuGet, verifique se há `Assembly Version Validation` erros exibidos no console do Unity. Esses erros ocorrem se os redirecionamentos de vinculação para assemblies com nomes fortes nos NuGet pacotes não estiverem sendo resolvidos corretamente para caminhos dentro do seu projeto Unity. Para resolver esse problema, configure a validação da versão de montagem do Unity:

1. No editor do Unity, acesse o menu principal e selecione Editar, Configurações do projeto e abra a seção Player.
2. Desmarque a opção Validação da versão de montagem.

Etapa 2: Adicionar o plug-in e os pacotes do SDK do servidor C#

1. Descompacte o GameLift plug-in da Amazon para download do Unity, que contém os dois pacotes.
2. Com seu projeto aberto no Unity Editor, vá até o menu principal e selecione Window, Package Manager.
3. Escolha o botão + para adicionar um novo pacote. Escolha a opção Adicionar pacote do tarball.
4. Em Selecionar pacotes no disco, localize o plug-in Amazon GameLift C# Server SDK para baixar arquivos do Unity e escolha o `com.amazonaws.gameliftserver.sdk-<version>.tgz` arquivo. Escolha Abrir para instalar o plug-in.
5. Em Selecionar pacotes no disco, localize o plug-in GameLift autônomo da Amazon para arquivos de download do Unity e escolha o `arquivocom.amazonaws.gamelift-<version>.tgz`. Escolha Abrir para instalar o plug-in.
6. Verifique se o plug-in independente foi adicionado ao seu projeto. Retorne à janela do editor Unity. Verifique o menu principal para ver o novo botão de GameLift menu da Amazon.

Etapa 3: importar o jogo de amostra (opcional)

O plug-in para Unity vem com um conjunto de amostras de ativos do jogo, incluindo cenas, que você pode adicionar ao seu projeto de jogo. A importação do jogo de amostra oferece um caminho rápido para testar, criar e implantar um jogo multijogador simples com a Amazon. GameLift O jogo de amostra já está totalmente integrado aos GameLift SDKs da Amazon, então você pode pular as tarefas de integração e concluir as tarefas restantes do fluxo de trabalho.

Ao usar o jogo de amostra, você pode configurar e ingressar em uma frota Amazon GameLift Anywhere hospedada localmente em apenas alguns minutos. Você pode implantar o jogo na Amazon GameLift e participar de um jogo ao vivo hospedado na nuvem em menos de uma hora.

Para importar o jogo de amostra:

1. Com seu projeto de jogo aberto no Unity Editor, acesse o GameLift menu Amazon e selecione Sample Game, Import Sample Game.
2. Depois que os arquivos forem importados, acesse o GameLift menu Amazon novamente e selecione Sample Game, Initialize Settings. Esta etapa configura seu projeto para criar o cliente e o servidor do jogo.

Quando a instalação estiver concluída, você verá duas novas cenas adicionadas ao seu projeto de jogo. Você também verá alguns ativos adicionais do projeto, incluindo um `GameLiftClientSettingsativo`.

Para obter mais detalhes sobre a interface de usuário e a jogabilidade da amostra, consulte a amostra do `readme` do jogo.

Configurar um perfil de usuário da AWS

Depois de instalar o plug-in, configure um perfil e vincule-o a uma conta da AWS de usuário válida. É possível manter vários perfis, mas só pode ter um perfil ativo de cada vez. Sempre que você trabalha no plug-in, selecione um perfil para usar.

A manutenção de vários perfis permite alternar entre diferentes cenários de hospedagem. Por exemplo, você pode configurar perfis com as mesmas credenciais da AWS, mas com Regiões da AWS diferentes. Ou você poderá configurar perfis com contas da AWS diferentes ou com diferentes conjuntos de usuários/permisões.

Note

Se você instalou a AWS CLI em sua estação de trabalho e já tem um perfil configurado, o GameLift plug-in da Amazon pode detectá-la e a listará como um perfil existente. O plug-in seleciona automaticamente qualquer perfil nomeado `[default]`. Você poderá usar um perfil existente ou criar outro.

Para configurar seu AWS perfil

1. No menu principal do editor Unity, escolha Amazon GameLift e selecione Definir perfis de AWS conta. Essa ação abre a janela do plugin. Abra a página Perfis de AWS usuário.
2. Se o plug-in detectar um perfil existente, você não será solicitado a criar um. Selecione um perfil existente ou escolha Adicionar outro perfil para criar um novo.
3. Se o plug-in não detectar um perfil existente, ele solicitará que você crie um. Você pode criar um novo perfil usando uma conta da AWS nova ou existente.

Note

Você precisa usar o Console de Gerenciamento da AWS para criar uma conta da AWS e criar ou atualizar um usuário com o conjunto de permissões adequado.

Ao configurar um perfil, você precisa das seguintes informações:

- Uma conta da AWS. Se precisar criar uma nova conta da AWS, siga as instruções para criar a conta. Consulte [Criar uma conta da AWS](#) para obter mais detalhes.
 - Um usuário AWS da conta com permissões para usar a Amazon GameLift e outros AWS serviços necessários. Consulte [Configurar um Conta da AWS](#) para obter instruções sobre como configurar um usuário AWS Identity and Access Management (IAM) com GameLift permissões da Amazon.
 - Credenciais para seu usuário da AWS. Esse usuário também precisa de acesso programático com credenciais de longo prazo. Essas credenciais consistem em um ID de chave de acesso da AWS e uma chave secreta AWS Consulte [Obter suas chaves de acesso](#) para obter mais detalhes.
 - Região AWS. Essa é uma localização geográfica em que você deseja criar seus recursos de hospedagem da AWS. Durante o desenvolvimento, recomendamos usar uma região próxima à sua localização física para minimizar a latência. Confira a lista de [regiões da AWS com suporte](#).
4. Ao selecionar ou criar um perfil, verifique o status de bootstrap do perfil e tome as medidas necessárias. Todos os perfis devem ser inicializados para usar a funcionalidade do plug-in da Amazon GameLift .

Para inicializar seu perfil:

O bootstrapping designa um bucket do Amazon S3 para uso com o perfil de usuário selecionado. O Amazon S3 é um AWS serviço essencial para armazenamento de dados e objetos. O bucket usado para armazenar configurações de projetos, criar artefatos e outras dependências. Os buckets não são compartilhados entre outros perfis.

Note

O bootstrapping cria novos AWS recursos e pode gerar custos.

1. Ao visualizar seus perfis na janela do plugin Perfis de AWS usuário, selecione o perfil que você deseja usar. Uma mensagem de aviso será exibida se o perfil ainda não tiver sido inicializado.
2. Na seção Bootstrap seu perfil, selecione um perfil na lista suspensa e verifique o status do bootstrap. Se o status indicar que não existe um bucket, escolha o botão Perfil do Bootstrap. Você pode definir o nome do bucket como um novo nome de bucket, inserir um bucket existente ao qual você tenha acesso ou manter o nome gerado automaticamente.
3. Aguarde até que o status do bootstrap mude para "Active". Isso pode levar alguns minutos. Quando o status é "Ativo", você pode usar o perfil para trabalhar com os recursos do plugin

Configure seu jogo para testes locais com a Amazon GameLift Anywhere

Nesse fluxo de trabalho, você adiciona código de jogo de cliente e servidor para a GameLift funcionalidade da Amazon e usa o plug-in para designar sua estação de trabalho local como host de servidor de jogos de teste. Depois de concluir as tarefas de integração, use o plug-in para criar seus componentes de cliente e servidor do jogo.

Para iniciar o fluxo de trabalho do Amazon GameLift Anywhere:

- No menu principal do editor Unity, escolha Amazon GameLift e selecione Host with Anywhere. Essa ação abre a página do plugin para configurar seu jogo com uma Anywhere frota @. A página apresenta um processo de cinco etapas para integrar, criar e lançar os componentes do jogo.

Defina seu perfil

Escolha o perfil que você deseja usar ao seguir esse fluxo de trabalho. O perfil selecionado afeta todas as etapas do fluxo de trabalho. Todos os recursos que você cria estão associados à conta do perfil da AWS e são colocados na Região da AWS padrão do perfil. As permissões do usuário do perfil determinam seu acesso aos recursos e ações da AWS.

1. Selecione um perfil na lista suspensa de perfis disponíveis. Se você ainda não tem um perfil ou deseja criar um novo, acesse o GameLift menu Amazon e escolha Definir perfis de AWS conta.

2. Se o status do bootstrap não for “Ativo”, escolha o perfil do Bootstrap e espere que o status mude para “Ativo”.

Integre seu jogo com a Amazon GameLift

Note

Se você importou o jogo de amostra, pode pular essa etapa. Os ativos do jogo de amostra já têm o código de servidor e cliente necessário.

Nesta etapa do fluxo de trabalho, você atualiza o código do cliente e do servidor no seu projeto de jogo.

- * Os servidores de jogos devem ser capazes de se comunicar com o GameLift serviço da Amazon para receber solicitações para iniciar uma sessão de jogo, fornecer informações de conexão da sessão de jogo e relatar o status.
- Os clientes do jogo devem ser capazes de obter informações sobre sessões de jogo, entrar ou iniciar sessões de jogo e obter informações de conexão para entrar em um jogo.

Integre seu código de servidor

Se você estiver usando seu próprio projeto de jogo com cenas personalizadas, use o código de amostra fornecido para adicionar o código de servidor necessário ao seu projeto de jogo:

1. Nos arquivos do projeto do jogo, abra a `Assets/Scripts/Server` pasta. Se não existir, crie-o.
2. Vá para o GitHub repositório [aws/ amazon-gamelift-plugin-unity](https://github.com/aws/amazon-gamelift-plugin-unity) e abra o caminho. `Samples~/SampleGame/Assets/Scripts/Server`
3. Localize o arquivo `GameLiftServer .cs.` e copie-o para a pasta `Server` do seu projeto de jogo. Ao criar um executável de servidor, use esse arquivo como destino de compilação.

O código de amostra inclui esses elementos mínimos necessários, que usam o SDK do servidor Amazon GameLift C# (versão 5):

- Inicializa um cliente de GameLift API da Amazon. A `InitSDK()` chamada com os parâmetros do servidor é necessária para uma frota do Amazon GameLift Anywhere. Essas configurações são definidas automaticamente para uso no plug-in.
- Implementa as funções de retorno de chamada necessárias para responder às solicitações do GameLift serviço da Amazon, incluindo `OnStartGameSessionOnProcessTerminate`, e `onHealthCheck`
- Chamadas `ProcessReady()` com uma porta designada para notificar o GameLift serviço da Amazon quando o processo do servidor estiver pronto para hospedar sessões de jogo.

Se você quiser personalizar o código do servidor de amostra, consulte estes recursos:

- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)

Integre seu código de cliente

Se você estiver usando seu próprio projeto de jogo com cenas personalizadas, precisará integrar a funcionalidade básica ao seu cliente de jogo. Você também precisa adicionar elementos de interface para que os jogadores possam entrar e participar de uma sessão de jogo. Use as APIs GameLift de serviços da Amazon (no AWS SDK) para obter informações da sessão do jogo, criar novas sessões de jogo ou participar de sessões de jogos existentes,

Ao criar um cliente para testes locais com uma frota Anywhere, você pode adicionar chamadas diretas ao GameLift serviço da Amazon. Ao desenvolver seu jogo para hospedagem na nuvem, ou se planeja usar frotas Anywhere para hospedagem de produção, você precisará criar um serviço de back-end do lado do cliente para lidar com toda a comunicação entre os clientes do jogo e o serviço da Amazon. GameLift

Para integrar a Amazon GameLift ao seu código de cliente, use os seguintes recursos como guia.

- Integre o cliente com a `GameLiftCoreApi` classe no GitHub repositório `amazon-gamelift-plugin-unity aws/`. Essa classe fornece controles para autenticação do jogador e para recuperar informações da sessão do jogo.
- Veja exemplos de integrações de jogos, disponíveis no GitHub repositório `amazon-gamelift-plugin-unity aws/`,. `Samples~/SampleGame/Assets/Scripts/Client/GameLiftClient.cs`
- Siga as instruções em `Adicionar Amazon GameLift ao seu cliente de jogo Unity`.

Para clientes de jogos que se conectam a uma frota Anywhere, seu cliente de jogo precisa das seguintes informações. O plug-in atualiza automaticamente seu projeto de jogo para usar os recursos que você criou no plug-in.

- FleetId - O identificador exclusivo da sua frota Anywhere.
- FleetLocation - A localização personalizada da sua frota Anywhere.
- AwsRegion - A AWS região onde sua frota Anywhere está hospedada. Essa é a região que você definiu no seu perfil de usuário.
- ProfileName - Um perfil de AWS credenciais em sua máquina local que permite acesso ao AWS SDK para. GameLift O cliente do jogo usa essas credenciais para autenticar solicitações no serviço Amazon GameLift .

Note

O perfil de credenciais é gerado pelo plug-in e armazenado na máquina local. Como resultado, você deve executar o cliente na máquina local (ou em uma máquina com o mesmo perfil).

Conecte-se a uma frota em qualquer lugar

Nesta etapa, você designa uma frota do Anywhere para usar. Uma frota do Anywhere define uma coleção de recursos computacionais, que podem estar localizados em qualquer lugar, para hospedagem de servidores de jogos.

- Se a AWS conta que você está usando atualmente tiver frotas Anywhere existentes, abra o campo suspenso Nome da frota e escolha uma frota. Essa lista suspensa mostra apenas as frotas do Anywhere na região da AWS para o perfil de usuário atualmente ativo.
- Se não houver frotas existentes, ou se você quiser criar uma nova, escolha Criar nova frota em qualquer lugar e forneça um nome para a frota.

Depois de escolher uma frota Anywhere para seu projeto, a Amazon GameLift verifica se o status da frota está ativo e exibe o ID da frota. Você pode acompanhar o progresso dessa solicitação no registro de saída do editor Unity.

Registrar um computador

Nesta etapa, você registra sua estação de trabalho local como um recurso computacional na nova frota do Anywhere.

1. Insira um nome de computação para a sua máquina local. Se você adicionar mais de uma computação na frota, os nomes deverão ser exclusivos.
2. Escolha Registrar computação. Você poderá acompanhar o progresso dessa solicitação no log de saída do editor do Unreal.

O plug-in registra sua estação de trabalho local com o endereço IP definido como localhost (127.0.0.1). Essa configuração pressupõe que você executará o cliente e o servidor do jogo na mesma máquina.

Em resposta a essa ação, a Amazon GameLift verifica se pode se conectar à computação e retorna informações sobre a computação recém-registrada.

Inicie o jogo

Nesta etapa, você cria os componentes do jogo e os inicia para jogar. Complete as seguintes tarefas:

1. Configure seu cliente de jogo. Nesta etapa, você solicita que o plug-in atualize um `GameLiftClientSettings` ativo para seu projeto de jogo. O plug-in usa esse recurso para armazenar determinadas informações que seu cliente de jogo precisa para se conectar ao GameLift serviço da Amazon.
 - a. Se você não importou e inicializou o jogo de amostra, crie um novo `GameLiftClientSettings` ativo. No menu principal do editor Unity, escolha Assets, Create GameLift, Client Settings. Se você criar várias cópias `GameLiftClientSettings` em seu projeto, o plug-in detectará isso automaticamente e notificará qual ativo o plug-in atualizará.
 - b. Em Launch Game, escolha Configurar cliente: aplicar configurações em qualquer lugar. Essa ação atualiza as configurações do cliente do jogo para usar a frota Anywhere que você acabou de configurar.
2. Crie e execute seu cliente de jogo.
 - a. Crie um executável de cliente usando o processo de compilação padrão do Unity. Em Arquivo, Configurações de compilação, mude a plataforma para Windows, Mac, Linux. Se você importou o jogo de amostra e inicializou as configurações, a lista de compilações e o destino da compilação serão atualizados automaticamente.

- b. Inicie uma ou mais instâncias do executável do cliente de jogo recém-criado.
3. Inicie um servidor de jogos em sua frota Anywhere. Escolha Servidor: Inicie o servidor no editor. Essa tarefa inicia um servidor ativo ao qual seu cliente pode se conectar, desde que o editor Unity permaneça aberto.
4. Inicie ou participe de uma sessão de jogo. Nas instâncias do seu cliente de jogo, use a interface do usuário para unir cada cliente a uma sessão de jogo. A forma como você faz isso depende de como você adicionou funcionalidade ao cliente.

Se você estiver usando o cliente de jogo de amostra, ele tem as seguintes características:

- Um componente de login do jogador. Ao se conectar a um servidor de jogo em uma frota Anywhere, não há validação do jogador. Você pode inserir qualquer valor para entrar na sessão do jogo.
- Uma interface de usuário simples para entrar no jogo. Quando um cliente tenta entrar em um jogo, ele procura automaticamente por uma sessão de jogo ativa com um espaço de jogador disponível. Se nenhuma sessão de jogo estiver disponível, o cliente solicitará uma nova sessão de jogo. Se uma sessão de jogo estiver disponível, o cliente solicitará entrar na sessão de jogo disponível. Ao testar seu jogo com vários clientes simultâneos, o primeiro cliente inicia a sessão do jogo e os clientes restantes ingressam automaticamente na sessão de jogo existente.
- Sessões de jogo com slots para quatro jogadores. Você pode iniciar até quatro instâncias de clientes de jogos simultaneamente e elas entrarão na mesma sessão de jogo.

Inicie a partir de um servidor executável (opcional)

Você pode criar e iniciar o executável do seu servidor de jogos para testes em uma frota Anywhere.

1. Crie um servidor executável usando o processo de compilação padrão do Unity. Em Arquivo, Configurações de Compilação, mude a plataforma para Servidor Dedicado e crie.
2. Obtenha um token de autenticação de curto prazo chamando o comando AWS CLI [get-compute-auth-token](#) com seu ID de frota e AWS região do Anywhere. O ID da frota é exibido no Connect to an Anywhere Fleet quando você cria a frota. A AWS região é exibida em Defina seu perfil quando você seleciona seu perfil ativo.

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

3. Inicie o executável do servidor de jogos recém-criado a partir de uma linha de comando e passe um token de autenticação válido.

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

Implante seu jogo em hospedagem em nuvem com frotas de EC2 gerenciadas

Nesse fluxo de trabalho, você usa o plug-in para preparar seu jogo para hospedagem em recursos computacionais baseados em nuvem que são gerenciados pela Amazon. GameLift Você adiciona o código do jogo do cliente e do servidor para a GameLift funcionalidade da Amazon e, em seguida, carrega a versão do seu servidor no GameLift serviço Amazon para hospedagem. Quando esse fluxo de trabalho estiver concluído, você terá servidores de jogos em execução na nuvem e um cliente de jogos funcional que poderá se conectar a eles.

Para iniciar o fluxo de trabalho do Amazon EC2 GameLift gerenciado pela Amazon:

- No menu principal do editor Unity, escolha Amazon GameLift e selecione Host with Managed EC2. Esse fluxo de trabalho apresenta um processo de seis etapas para integrar, criar, implantar e iniciar os componentes do jogo.

Defina seu perfil

Escolha o perfil que você deseja usar ao seguir esse fluxo de trabalho. O perfil selecionado afeta todas as etapas do fluxo de trabalho. Todos os recursos que você cria estão associados à conta do perfil da AWS e são colocados na Região da AWS padrão do perfil. As permissões do usuário do perfil determinam seu acesso aos recursos e ações da AWS.

1. Selecione um perfil na lista suspensa de perfis disponíveis. Se você ainda não tem um perfil ou deseja criar um novo, acesse o GameLift menu Amazon e escolha Definir perfis de AWS conta.
2. Se o status do bootstrap não for “Ativo”, escolha o perfil do Bootstrap e espere que o status mude para “Ativo”.

Integre seu jogo com a Amazon GameLift

Para essa tarefa, você faz atualizações no código do cliente e do servidor em seu projeto de jogo.

- Os servidores de jogos devem ser capazes de se comunicar com o GameLift serviço da Amazon para receber solicitações para iniciar uma sessão de jogo, fornecer informações de conexão da sessão de jogo e relatar o status.
- Os clientes do jogo devem ser capazes de obter informações sobre sessões de jogo, entrar ou iniciar sessões de jogo e obter informações de conexão para entrar em um jogo.

Note

Se você importou o jogo de amostra, pode pular essa etapa. Os ativos do jogo de amostra já têm o código de servidor e cliente necessário.

Integre seu código de servidor

Ao usar seu próprio projeto de jogo com cenas personalizadas, use o código de amostra fornecido para adicionar o código de servidor necessário ao seu projeto de jogo. Se você integrou seu projeto de jogo para testes com uma frota Anywhere, você já concluiu as instruções nesta etapa.

1. Nos arquivos do projeto do jogo, abra a `Assets/Scripts/Server` pasta. Se não existir, crie-o.
2. Vá para o GitHub repositório [aws/ amazon-gamelift-plugin-unity](https://github.com/aws/amazon-gamelift-plugin-unity) e abra o caminho. `Samples~/SampleGame/Assets/Scripts/Server`
3. Localize o arquivo `GameLiftServer.cs` e copie-o para a `Server` pasta do seu projeto de jogo. Ao criar um executável de servidor, use esse arquivo como destino de compilação.

O código de amostra inclui esses elementos mínimos necessários, que usam o SDK do servidor Amazon GameLift C# (versão 5):

- Inicializa um cliente de GameLift API da Amazon. A chamada `initSDK()` com parâmetros de servidor é necessária para uma frota do Amazon Anywhere. GameLift Essas configurações são definidas automaticamente para uso no plug-in.
- Implementa as funções de retorno de chamada necessárias para responder às solicitações do GameLift serviço da Amazon, incluindo `OnStartGameSessionOnProcessTerminate`, e `onHealthCheck`
- Chamadas `ProcessReady()` com uma porta designada para notificar o GameLift serviço da Amazon quando o processo do servidor estiver pronto para hospedar sessões de jogo.

Se você quiser personalizar o código do servidor de amostra, consulte estes recursos:

- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)

Integre seu código de cliente

Para clientes de jogos que se conectam a servidores de jogos baseados em nuvem, é uma prática recomendada usar um serviço de back-end do lado do cliente para fazer chamadas para o serviço da Amazon GameLift, em vez de fazer as chamadas diretamente do cliente do jogo.

No fluxo de trabalho do plug-in para hospedagem em uma frota EC2 gerenciada, cada cenário de implantação inclui um serviço de back-end pré-criado que inclui os seguintes componentes:

- Um conjunto de funções do Lambda e tabelas do DynamoDB que são usadas para solicitar sessões de jogo e recuperar informações da sessão de jogo. Esses componentes usam um gateway de API como proxy.
- Um grupo de usuários do Amazon Cognito que gera IDs de jogadores exclusivos e autentica conexões de jogadores.

Para usar esses componentes, seu cliente de jogo precisa da funcionalidade de enviar solicitações ao serviço de back-end para fazer o seguinte:

- Crie um usuário jogador no grupo de usuários do AWS Cognito e autentique-o.
- Participe de uma sessão de jogo e receba informações de conexão.
- Entre em um jogo usando matchmaking.

Use os seguintes recursos como guia.

- Integre o cliente com a [GameLiftCoreApi](#) classe no GitHub repositório [amazon-gamelift-plugin-unityaws/](#). Essa classe fornece controles para autenticação do jogador e para recuperar informações da sessão do jogo.
- Para ver os exemplos de integrações de jogos, acesse o GitHub repositório [amazon-gamelift-plugin-unityaws/](#),. `Samples~/SampleGame/Assets/Scripts/Client/GameLiftClient.cs`
- [Adicione GameLift a Amazon ao seu cliente de jogos Unity](#).

Selecione o cenário de implantação

Nesta etapa, você escolhe a solução de hospedagem de jogos que deseja implantar no momento. Você poderá ter várias implantações do seu jogo, usando qualquer um dos cenários.

- **Frota de região única:** implanta seu servidor de jogo em uma única frota de recursos de hospedagem na região padrão AWS do perfil ativo. Esse cenário é um bom ponto de partida para testar a integração do servidor da AWS e a configuração de compilação do servidor. Ele implanta os seguintes recursos:
 - frota (sob demanda) da AWS com a versão do seu servidor de jogos instalada e em execução.
 - Grupo de usuários e cliente do Amazon Cognito para permitir que os jogadores se autentiquem e iniciem um jogo.
 - Autorizador de gateway de API que vincula o grupo de usuários às APIs.
 - WebACL para controle de utilização excessiva de jogadores para o gateway da API.
 - Gateway de API + função do Lambda para os jogadores solicitarem um slot de jogo. Essa função é chamada `CreateGameSession()` se nenhuma estiver disponível.
 - Gateway de API + função do Lambda para que os jogadores obtenham informações de conexão para sua solicitação de jogo.
- **FlexMatch frota:** implanta seu servidor de jogo em um conjunto de frotas e configura um FlexMatch matchmaker com regras para criar partidas de jogadores. Esse cenário usa hospedagem spot de baixo custo com uma estrutura de várias frotas e vários locais para uma disponibilidade durável. Essa abordagem é útil quando você está pronto para começar a projetar um componente matchmaker para sua solução de hospedagem. Nesse cenário, você criará os recursos básicos para essa solução, que poderão ser personalizados posteriormente, conforme necessário. Ele implanta os seguintes recursos:
 - FlexMatch configuração de matchmaking e regras de matchmaking definidas para aceitar solicitações de jogadores e formar partidas.
 - Três frotas da AWS com sua versão de servidor de jogo instalada e funcionando em vários locais. Inclui duas frotas spot e uma frota sob demanda como backup.
 - Fila de posicionamento de sessões de jogo da AWS que atende às solicitações de partidas propostas, encontrando o melhor recurso de hospedagem possível (com base na viabilidade, custo, latência do jogador etc.) e iniciando uma sessão de jogo.
 - Grupo de usuários e cliente do Amazon Cognito para permitir que os jogadores se autentiquem e iniciem um jogo.
 - **Autorizador de gateway de API que vincula o grupo de usuários às APIs.**

- WebACL para controle de utilização excessiva de jogadores para o gateway da API.
- Gateway de API + função do Lambda para os jogadores solicitarem um slot de jogo. Esta função chama `StartMatchmaking()`.
- Gateway de API + função do Lambda para que os jogadores obtenham informações de conexão para sua solicitação de jogo.
- Tabelas do Amazon DynamoDB para armazenar ingressos de marcação de jogos para jogadores e informações da sessão do jogo.
- Tópico do SNS + Função Lambda para `GameSessionQueue` lidar com eventos.

Defina os parâmetros do jogo

Nesta etapa, você descreve seu jogo para o qual fazer o upload da AWS.

- Nome do jogo: forneça um nome significativo para seu projeto de jogo. Esse nome é usado dentro do plugin.
- Nome da frota: forneça um nome significativo para sua frota EC2 gerenciada. A Amazon GameLift usa esse nome (junto com o ID da frota) ao listar recursos no AWS console.
- Nome da compilação: forneça um nome significativo para a compilação do seu servidor. AWS usa esse nome para se referir à cópia da versão do seu servidor que é enviada para a Amazon GameLift e usada para implantações.
- Parâmetros de inicialização: insira instruções opcionais para execução ao iniciar o executável do servidor em uma instância de frota EC2 gerenciada. O tamanho máximo é de 1024 caracteres.
- Pasta do servidor do jogo: forneça o caminho para uma pasta local contendo a compilação do seu servidor.
- Arquivo do servidor do jogo: especifique o nome do arquivo executável do servidor.

Cenário de implantação

Nesta etapa, você implanta seu jogo em uma solução de hospedagem em nuvem com base no cenário de implantação escolhido. Esse processo pode levar até 40 minutos enquanto a AWS valida a criação do servidor, provisiona recursos de hospedagem, instala o servidor do jogo, inicia os processos do servidor e os prepara para hospedar sessões de jogo.

Para iniciar a implantação, escolha `Implantar CloudFormation`. É possível acompanhar o status de sua hospedagem de jogos aqui. Para obter informações mais detalhadas, você pode entrar na AWS

em console de gerenciamento da AWS e visualizar notificações de eventos. Certifique-se de fazer login usando a mesma conta, usuário e região da AWS, como perfil de usuário ativo no plug-in.

Quando a implantação estiver concluída, você terá seu servidor de jogos instalado em uma instância do EC2 da AWS. Pelo menos um processo do servidor está em execução e pronto para iniciar uma sessão de jogo.

Inicie o cliente do jogo

Quando sua frota é implantada com sucesso, agora você tem servidores de jogos em execução e disponíveis para hospedar sessões de jogo. Agora você pode criar seu cliente, iniciá-lo, conectar-se para participar da sessão do jogo.

1. Configure seu cliente de jogo. Nesta etapa, você solicita que o plug-in atualize um `GameLiftClientSettings` ativo para seu projeto de jogo. O plug-in usa esse recurso para armazenar determinadas informações que seu cliente de jogo precisa para se conectar ao GameLift serviço da Amazon.
 - a. Se você não importou e inicializou o jogo de amostra, crie um novo `GameLiftClientSettings` ativo. No menu principal do editor Unity, escolha Assets, Create GameLift, Client Settings. Se você criar várias cópias `GameLiftClientSettings` em seu projeto, o plug-in detectará isso automaticamente e notificará qual ativo o plug-in atualizará.
 - b. Em Launch Game, escolha Configure Client: Apply Managed EC2 Settings. Essa ação atualiza as configurações do seu cliente de jogo para usar a frota EC2 gerenciada que você acabou de implantar.
2. Crie seu cliente de jogo. Crie um executável de cliente usando o processo de compilação padrão do Unity. Em Arquivo, Configurações de compilação, mude a plataforma para Windows, Mac, Linux. Se você importou o jogo de amostra e inicializou as configurações, a lista de compilações e o destino da compilação serão atualizados automaticamente.
3. Inicie o executável do cliente de jogo recém-criado. Para começar a jogar, inicie de duas a quatro instâncias do cliente e use a interface de usuário em cada uma para participar de uma sessão de jogo.

Se você estiver usando o cliente de jogo de amostra, ele tem as seguintes características:

- Um componente de login do jogador. Ao se conectar a um servidor de jogo em uma frota Anywhere, não há validação do jogador. Você pode inserir qualquer valor para entrar na sessão do jogo.

- Uma interface de usuário simples para entrar no jogo. Quando um cliente tenta entrar em um jogo, ele procura automaticamente por uma sessão de jogo ativa com um espaço de jogador disponível. Se nenhuma sessão de jogo estiver disponível, o cliente solicitará uma nova sessão de jogo. Se uma sessão de jogo estiver disponível, o cliente solicitará entrar na sessão de jogo disponível. Ao testar seu jogo com vários clientes simultâneos, o primeiro cliente inicia a sessão do jogo e os clientes restantes ingressam automaticamente na sessão de jogo existente.
- Sessões de jogo com slots para quatro jogadores. Você pode iniciar até quatro instâncias de clientes de jogos simultaneamente e elas entrarão na mesma sessão de jogo.

Guia do GameLift plugin Amazon para Unity para servidor SDK 4.x

Note

Este tópico fornece informações para uma versão anterior do GameLift plug-in da Amazon para Unity. A versão 1.0.0 (lançada em 2021) usa o Amazon GameLift Server SDK 4.x ou anterior. Para obter a documentação sobre a versão mais recente do plug-in, que usa o SDK 5.x do servidor e é compatível com a Amazon GameLift Anywhere, consulte. [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#)

GameLift A Amazon fornece ferramentas para preparar seus servidores de jogos multijogador para serem executados na Amazon GameLift. O GameLift plug-in da Amazon para Unity facilita a integração da Amazon GameLift em seus projetos de jogos da Unity e a implantação de GameLift recursos da Amazon para hospedagem na nuvem. Use o plug-in do Unity para acessar as GameLift APIs da Amazon e implantar AWS CloudFormation modelos para cenários de jogos comuns.

Depois de configurar o plug-in, você pode experimentar a [amostra do Amazon GameLift Unity](#) em GitHub.

Tópicos

- [Integre a Amazon GameLift com um projeto de servidor de jogos Unity](#)
- [Integre a Amazon GameLift com um projeto de cliente de jogo Unity](#)
- [Instale e configure o plug-in](#)
- [Teste seu jogo localmente](#)
- [Implemente um cenário](#)

- [Integre jogos com a Amazon GameLift no Unity](#)
- [Importe e execute um jogo de amostra](#)

Integre a Amazon GameLift com um projeto de servidor de jogos Unity

Note

Este tópico fornece informações para uma versão anterior do GameLift plug-in da Amazon para Unity. A versão 1.0.0 (lançada em 2021) usa o Amazon GameLift Server SDK 4.x ou anterior. Para obter a documentação sobre a versão mais recente do plug-in, que usa o SDK 5.x do servidor e é compatível com a Amazon GameLift Anywhere, consulte. [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#)

Este tópico ajuda você a preparar seu servidor de jogos personalizado para hospedagem na Amazon GameLift. O servidor do jogo deve ser capaz de notificar a Amazon GameLift sobre seu status, iniciar e interromper sessões de jogo quando solicitado e realizar outras tarefas. Para ter mais informações, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Pré-requisitos

Antes de integrar o servidor de jogos, execute as tarefas a seguir:

- [Configurar uma função de serviço do IAM para a Amazon GameLift](#)
- [Instale o plug-in para Unity](#)

Configure um novo processo de servidor

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Configure a comunicação com a Amazon GameLift e informe que o processo do servidor está pronto para hospedar uma sessão de jogo.

1. Inicialize o SDK do servidor ao chamar `InitSDK()`.

2. Para preparar o servidor para aceitar uma sessão de jogo, chame `ProcessReady()` com a porta de conexão e os detalhes da localização da sessão de jogo. Inclua os nomes das funções de retorno de chamada que o GameLift serviço da Amazon invoca, como, `OnGameSession()`, `OnGameSessionUpdate()`. `OnProcessTerminate()` `OnHealthCheck()` A Amazon GameLift pode levar alguns minutos para fornecer um retorno de chamada.
3. A Amazon GameLift atualiza o status do processo do servidor para `ACTIVE`.
4. A Amazon GameLift liga `onHealthCheck` periodicamente.

O exemplo de código a seguir mostra como configurar um processo de servidor simples com a Amazon GameLift.

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

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

var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);

// Implement callback functions
void OnGameSession(GameSession gameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    // When ready to receive players
    var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}
```

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

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

Iniciar uma sessão de jogo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Depois que a inicialização do jogo estiver concluída, você poderá iniciar uma sessão de jogo.

1. Implemente a função de retorno de chamada `onStartGameSession`. A Amazon GameLift invoca esse método para iniciar uma nova sessão de jogo no processo do servidor e receber conexões de jogadores.
2. Para ativar uma sessão de jogo, chame `ActivateGameSession()`. Para obter mais informações sobre o SDK, consulte [Referência do SDK do servidor do Amazon GameLift \(C#\): Ações](#).

O exemplo de código a seguir ilustra como iniciar uma sessão de jogo com a Amazon GameLift.

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

Encerrar uma sessão de jogo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Notifique a Amazon GameLift quando uma sessão de jogo estiver terminando. Como melhor prática, desligue os processos do servidor após a conclusão das sessões de jogo para reciclar e atualizar os recursos de hospedagem.

1. Configure uma função chamada `onProcessTerminate` para receber solicitações da Amazon GameLift e `ligarProcessEnding()`.
2. O status do processo muda para `TERMINATED`.

O exemplo a seguir descreve como finalizar um processo para uma sessão de jogo.

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

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

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

Crie servidores, construa e faça o upload para a Amazon GameLift

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Depois de integrar seu servidor de jogos com a Amazon GameLift, faça o upload dos arquivos de compilação para uma frota para que a Amazon GameLift possa implantá-los para hospedagem de jogos. Para obter mais informações sobre como fazer o upload do seu servidor para a Amazon

GameLift, consulte [Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift](#).

Integre a Amazon GameLift com um projeto de cliente de jogo Unity

Note

Este tópico fornece informações para uma versão anterior do GameLift plug-in da Amazon para Unity. A versão 1.0.0 (lançada em 2021) usa o Amazon GameLift Server SDK 4.x ou anterior. Para obter a documentação sobre a versão mais recente do plug-in, que usa o SDK 5.x do servidor e é compatível com a Amazon GameLift Anywhere, consulte [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#)

Este tópico ajuda você a configurar um cliente de jogo para se conectar às sessões de jogos GameLift hospedadas pela Amazon por meio de um serviço de back-end. Use as GameLift APIs da Amazon para iniciar matchmaking, solicitar posicionamento em sessões de jogo e muito mais.

Adicione código ao projeto de serviço de back-end para permitir a comunicação com o GameLift serviço da Amazon. Um serviço de back-end gerencia toda a comunicação do cliente do jogo com o GameLift serviço. Para obter mais informações sobre esses serviços de back-end, consulte [Projete seu serviço de cliente de jogos](#).

Um servidor de back-end executa as seguintes tarefas do cliente do jogo:

- Personalize a autenticação para seus jogadores.
- Solicite informações sobre sessões de jogo ativas no GameLift serviço da Amazon.
- Cria uma sessão de jogo.
- Adicionar um jogador a uma sessão de jogo existente.
- Remover um jogador de uma sessão de jogo existente.

Tópicos

- [Pré-requisitos](#)
- [Inicializar um cliente de jogo](#)
- [Crie a sessão do jogo em uma frota específica](#)
- [Adicionar jogadores às sessões de jogo](#)
- [Remova um jogador de uma sessão de jogo](#)

Pré-requisitos

Antes de configurar a comunicação do servidor do jogo com o GameLift cliente da Amazon, conclua as seguintes tarefas:

- [Configurar um Conta da AWS](#)
- [Instale o plug-in para Unity](#)
- [Integre a Amazon GameLift com um projeto de servidor de jogos Unity](#)
- [Configurar as frotas do Amazon GameLift](#)

Inicializar um cliente de jogo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Adicione código para inicializar um cliente de jogo. Execute esse código no lançamento, é necessário para outras GameLift funções da Amazon.

1. Inicializar `AmazonGameLiftClient`. Chame `AmazonGameLiftClient` com uma configuração de cliente padrão ou uma configuração personalizada. Para obter mais informações sobre como configurar um cliente, consulte [Configure a Amazon GameLift em um serviço de back-end](#).
2. Gere um ID de jogador exclusivo para cada jogador se conectar a uma sessão de jogo. Para obter mais informações, consulte [Gerar IDs de jogador](#).

Os exemplos a seguir mostram como configurar um GameLift cliente da Amazon.

```
public class GameLiftClient
{
    private GameLift gl;
    //A sample way to generate random player IDs.
    bool includeBrackets = false;
    bool includeDashes = true;
    string playerId = AZ::Uuid::CreateRandom().ToString<string>(includeBrackets,
includeDashes);
}
```

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

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

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

Crie a sessão do jogo em uma frota específica

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Adicione um código para iniciar novas sessões de jogos nas frotas implantadas e disponibilizá-las para jogadores. Depois GameLift que a Amazon criar a nova sessão de jogo e retornar uma `GameSession`, você poderá adicionar jogadores a ela.

- Faça uma solicitação para uma nova sessão de jogo.
 - Se seu jogo usa frotas, chame `CreateGameSession()` com um ID de frota ou alias, um nome de sessão e o número máximo de jogadores simultâneos no jogo.
 - Se o seu jogo usa filas, chame `StartGameSessionPlacement()`.

O exemplo a seguir mostra como criar uma sessão de jogo.


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

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

Adicionar jogadores às sessões de jogo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Depois GameLift que a Amazon criar a nova sessão de jogo e devolver um `GameSession` objeto, você poderá adicionar jogadores a ela.

1. Reserve um slot de jogador em uma sessão de jogo criando uma nova sessão de jogador. Use `CreatePlayerSession` ou `CreatePlayerSessions` com o ID da sessão do jogo e um ID exclusivo para cada jogador.
2. Conecte-se à sessão do jogo. Recupere o objeto `PlayerSession` para obter as informações de conexão da sessão do jogo. É possível usar essas informações para estabelecer uma conexão direta com o processo do servidor:
 - a. Use a porta especificada e o nome DNS ou endereço IP do processo do servidor.

- b. Use o nome DNS e a porta de suas frotas. O nome DNS e a porta serão necessários se suas frotas tiverem a geração de certificados TLS habilitada.
- c. Faça referência ao ID da sessão do jogador. O ID da sessão do jogador é necessário se o servidor do jogo validar as conexões de entrada do jogador.

Os exemplos a seguir demonstram como reservar uma vaga de jogador em uma sessão de jogo.

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

O código a seguir ilustra como conectar um jogador à sessão do jogo.

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

Remova um jogador de uma sessão de jogo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Você poderá remover os jogadores da sessão de jogo quando eles saírem do jogo.

1. Notifique o GameLift serviço da Amazon de que um jogador se desconectou do processo do servidor. Chame `RemovePlayerSession` com o ID da sessão do jogador.
2. Verifique se isso `RemovePlayerSession` retorna `Success`. Em seguida, a Amazon GameLift altera a vaga do jogador para ficar disponível, que a Amazon GameLift pode atribuir a um novo jogador.

O exemplo a seguir ilustra como remover uma sessão de jogador.

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

Instale e configure o plug-in

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Esta seção descreve como baixar, instalar e configurar o GameLift plug-in da Amazon para Unity, versão 1.0.0.

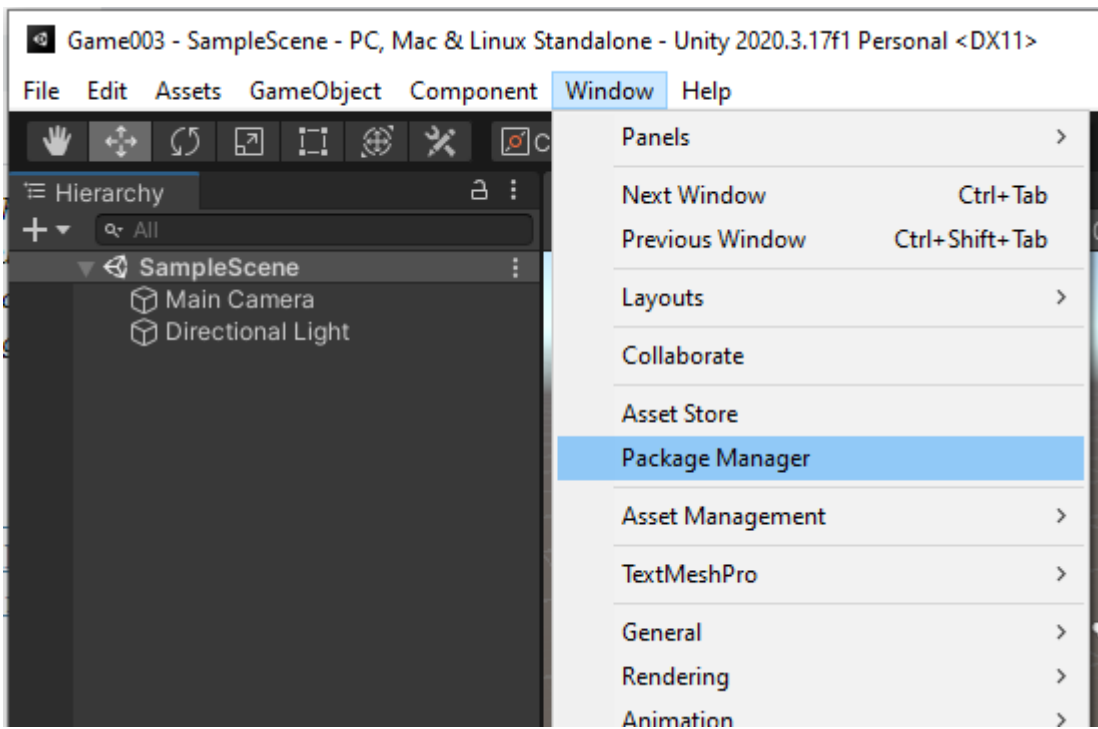
Pré-requisitos

- Unity para Windows 2019.4 LTS, Windows 2020.3 LTS ou Unity para macOS
- Versão atual do Java

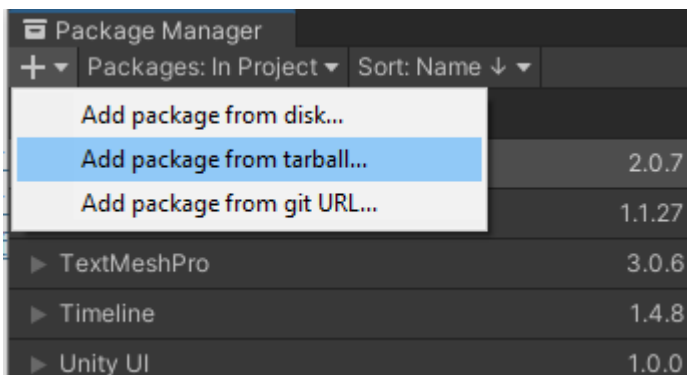
- Versão atual do.NET 4.x

Para fazer download e instalar o plug-in para Unity

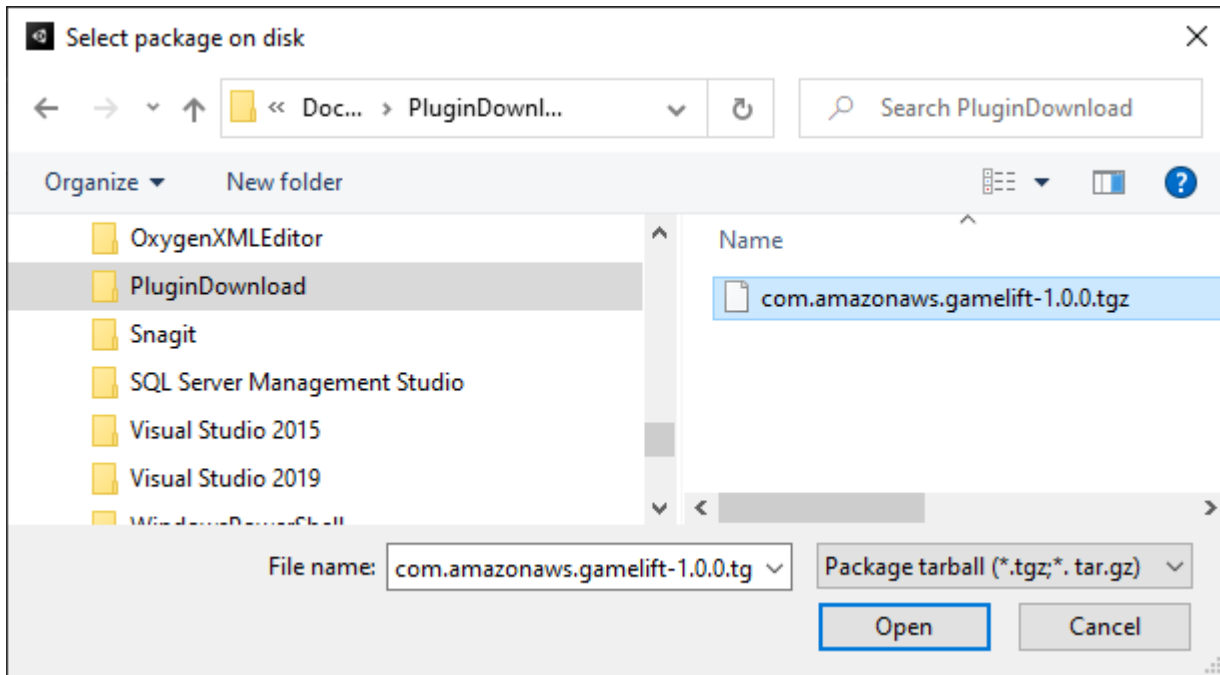
1. Baixe o GameLift plug-in da Amazon para Unity. Você pode encontrar a versão mais recente na página do [repositório do GameLift plug-in Amazon para Unity](#). Na [versão mais recente](#), escolha Ativos e, em seguida, baixe o arquivo com `.amazonaws.gamelift-version.tgz`.
2. Inicie o Unity e escolha um projeto.
3. Na barra de navegação superior, em Window, escolha Gerenciador de pacotes:



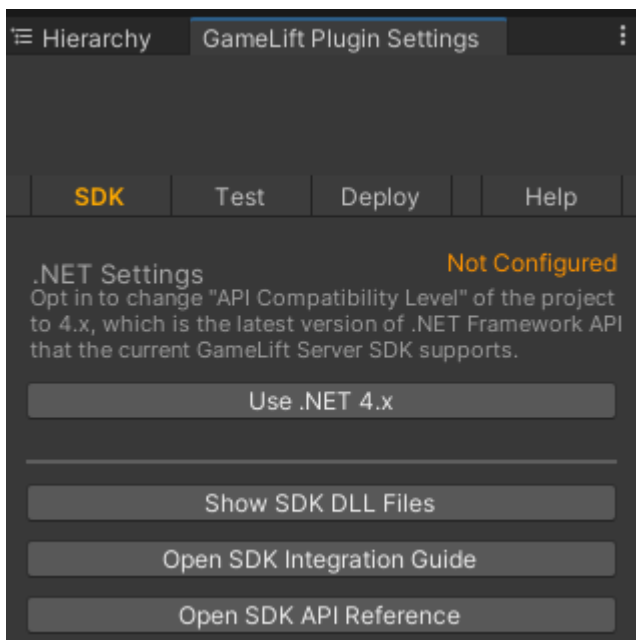
4. No guia Gerenciador de pacotes, escolha + e, em seguida, escolha Adicionar pacote do tarball...:



- Na janela Selecionar pacotes no disco, navegue até a pasta com `.amazonaws.gamelift`, escolha o arquivo com `.amazonaws.gamelift-version.tgz` e escolha Abrir:



- Depois que o Unity carrega o plug-in, a Amazon GameLift aparece como um novo item no menu Unity. Pode demorar alguns minutos para instalar e recompilar scripts. A guia Configurações do GameLift plug-in da Amazon é aberta automaticamente.



- No painel SDK, escolha Usar o .NET 4.x.

Quando configurado, o status muda de Não configurado para Configurado.

Teste seu jogo localmente

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Use o Amazon GameLift Local para executar a Amazon GameLift em seu dispositivo local. Você pode usar o Amazon GameLift Local para verificar alterações no código em segundos, sem uma conexão de rede.

Configurar testes locais

1. Na janela do plug-in para Unity, escolha o guia Teste.
2. No painel Teste, escolha Baixar Amazon GameLift Local. O plug-in para Unity abre uma janela do navegador e baixa o arquivo `GameLift_06_03_2021.zip` para sua pasta de downloads.

O download inclui o SDK do servidor C#, arquivos de origem do .NET e um componente do .NET compatível com o Unity.

3. Não descompacte o arquivo `GameLift_06_03_2021.zip` obtido por download.
4. Na janela Amazon GameLift Plugin Settings, escolha Amazon GameLift Local Path, navegue até a pasta descompactada, escolha o arquivo **GameLiftLocal.jar** e escolha Abrir.

Quando configurado, o status do teste local muda de Não configurado para Configurado.

5. Verifique o status do JRE. Se o status for Não configurado, escolha Baixar o JRE e instale a versão recomendada do Java.

Depois de instalar e configurar o ambiente Java, o status muda para Configurado.

Execute seu jogo local

1. No guia Plug-in para Unity, escolha o guia Teste.
2. No painel Teste, escolha Abrir interface de teste local.
3. Na janela Teste local, especifique um Caminho executável do servidor. Selecione... para selecionar o caminho e o nome do executável do seu aplicativo de servidor.
4. Na janela Teste local, especifique uma porta local GL.

5. Escolha Implantar e executar para implantar e executar o servidor.
6. Para parar seu servidor de jogos, escolha Parar ou feche as janelas do servidor de jogos.

Implemente um cenário

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Um cenário usa um AWS CloudFormation modelo para criar os recursos necessários para implantar uma solução de hospedagem em nuvem para seu jogo. Esta seção descreve os cenários que a Amazon GameLift oferece e como usá-los.

Pré-requisitos

Para implantar o cenário, você precisa de uma função do IAM para o GameLift serviço da Amazon. Para obter informações sobre como criar uma função para a Amazon GameLift, consulte [Configurar um Conta da AWS](#).

Cada cenário exige permissões para os seguintes recursos:

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

Cenários

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

O GameLift plug-in da Amazon para Unity inclui os seguintes cenários:

Somente Autenticação

Esse cenário cria um serviço de back-end de jogos que realiza a autenticação do jogador sem a capacidade do servidor de jogos. O modelo cria os seguintes recursos na sua conta:

- Um grupo de usuários do Amazon Cognito para armazenar informações de autenticação de jogadores.
- Um processador AWS Lambda baseado em endpoint REST do Amazon API Gateway que inicia jogos e visualiza as informações de conexão do jogo.

Frota de uma única região

Esse cenário cria um serviço de back-end de jogos com uma única GameLift frota da Amazon. Ele cria os recursos a seguir:

- Um grupo de usuários do Amazon Cognito para que um jogador se autentique e inicie um jogo.
- Um processador AWS Lambda para pesquisar uma sessão de jogo existente com uma vaga de jogador aberta na frota. Se não conseguir encontrar um slot aberto, ele cria uma nova sessão de jogo.

Frota multirregional com fila e marcador de jogos personalizado

Esse cenário forma partidas usando GameLift filas da Amazon e um matchmaker personalizado para agrupar os jogadores mais velhos no grupo de espera. Ele cria os recursos a seguir:

- Um tópico do Amazon Simple Notification Service para o qual a Amazon GameLift publica mensagens. Para obter mais informações sobre tópicos e notificações do SNS, consulte [Configurar notificações de eventos para o posicionamento de sessões de jogo](#).

- Uma função do Lambda que é invocada pela mensagem que comunica os detalhes do posicionamento e da conexão do jogo.
- Uma tabela do Amazon DynamoDB para armazenar detalhes de posicionamento e conexão do jogo. As chamadas `GetGameConnection` são lidas dessa tabela e retornam as informações de conexão para o cliente do jogo.

Identifique frotas com uma fila e um marcador de jogos personalizado

Esse cenário forma correspondências usando GameLift filas da Amazon e um matchmaker personalizado e configura três frotas. Ele cria os recursos a seguir:

- Duas frotas Spot que contêm diferentes tipos de instâncias para fornecer durabilidade em caso de indisponibilidade do Spot.
- Uma frota sob demanda que atua como backup para as outras frotas Spot. Para obter mais informações sobre como projetar suas frotas, consulte [Guia de criação de frota do Amazon GameLift](#).
- Uma GameLift fila da Amazon para manter a disponibilidade do servidor alta e o custo baixo. Para obter mais informações e melhores práticas sobre filas, consulte [Criar uma fila de sessões de jogo](#).

FlexMatch

Este cenário usa FlexMatch, um serviço gerenciado de matchmaking, para combinar jogadores. Para obter mais informações sobre FlexMatch, consulte [O que é a Amazon GameLift FlexMatch](#). Este cenário cria os recursos a seguir:

- Uma função do Lambda para criar um ticket de marcador de jogos depois de receber solicitações `StartGame`.
- Uma função Lambda separada para ouvir os eventos da FlexMatch partida.

Para evitar cobranças desnecessárias da sua Conta da AWS, remova os recursos criados por cada cenário depois de terminar de usá-los. Exclua a pilha AWS CloudFormation correspondente.

Atualizar AWS credenciais

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

O GameLift plug-in da Amazon para Unity exige credenciais de segurança para implantar um cenário. Você poderá criar novas credenciais ou usar credenciais existentes.

Para obter mais informações sobre como configurar credenciais, consulte [Noções básicas e obter suas credenciais da AWS](#).

Para atualizar as credenciais do AWS

1. No Unity, no guia Plug-in para Unity, escolha o guia Implantar.
2. No painel Implantar, escolha Credenciais da AWS.
3. Você poderá criar novas credenciais da AWS ou escolher credenciais existentes.
 - Para criar credenciais, escolha Criar novo perfil de credenciais e, em seguida, especifique o Nome do novo perfil, ID da chave de acesso da AWS, chave secreta da AWS e Região da AWS.
 - Para escolher as credenciais existentes, escolha Escolher perfil de credenciais existente e, em seguida, selecione um nome de perfil e Região da AWS.
4. Na janela Atualizar Credenciais da AWS, escolha Atualizar Perfil de Credenciais.

Atualizar o bootstrap da conta

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.


A localização do bootstrap é um bucket do Amazon S3 usado durante a implantação. Ele é usado para armazenar ativos do servidor de jogos e outras dependências. A Região da AWS que você escolher para o bucket deve ser a mesma região que você usará para a implantação do cenário.

Para obter mais informações sobre os buckets do Amazon S3, consulte [Criar, configurar e trabalhar com os buckets do Amazon Simple Storage Service](#).

Para atualizar a localização do bootstrap da conta

1. No Unity, no guia Plug-in para Unity, escolha a guia Implantar.
2. No painel Implantar, escolha Atualizar conta Bootstrap.
3. Na janela Account Bootstrapping, você escolhe um bucket do Amazon S3 existente ou cria um novo bucket do Amazon S3:
 - Para escolher um bucket existente, escolha Escolher bucket do Amazon S3 existente e Atualizar para salvar sua seleção.
 - Escolha Criar o bucket do Amazon S3 para criar um bucket do Amazon Simple Storage Service e, em seguida, escolha uma política. A política especifica quando o bucket do Amazon S3 expirará. Escolha Criar para criar o bucket.

Implemente um cenário de jogo

 Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

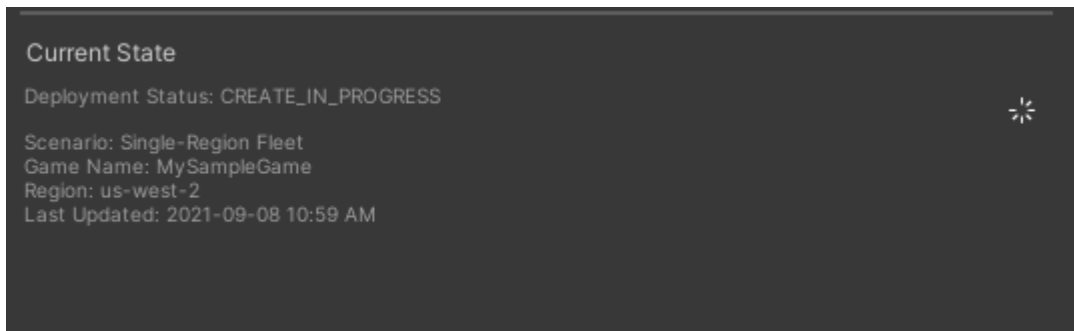
Você pode usar um cenário para testar seu jogo com a Amazon GameLift. Cada cenário usa um modelo AWS CloudFormation para criar uma pilha com os recursos necessários. A maioria dos cenários exige um servidor de jogo executável e um caminho de construção. Quando você implanta o cenário, a Amazon GameLift copia os ativos do jogo para o local de bootstrap como parte da implantação.

Você deve configurar as credenciais da AWS e um bootstrap de conta da AWS, para implantar um cenário.

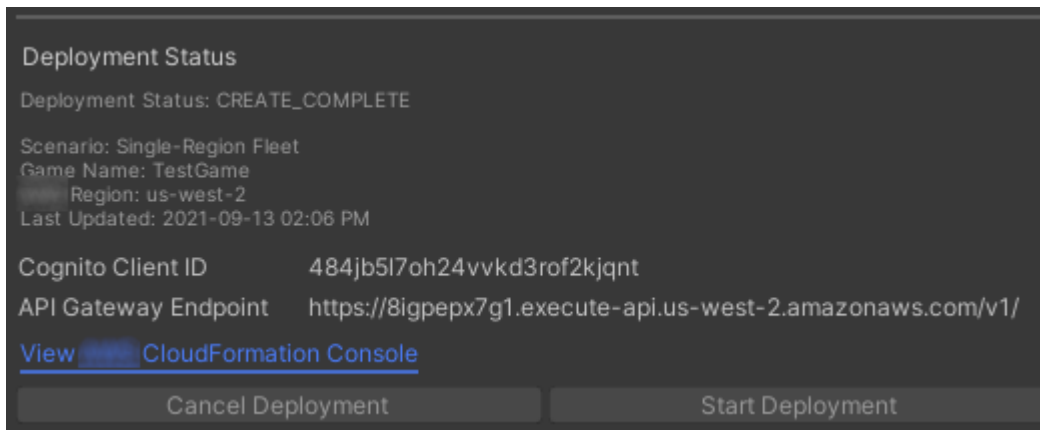
Para implantar um cenário

1. No Unity, no guia Plug-in para Unity, escolha o guia Implantar.
2. No painel Implantar, escolha Abrir interface de implantação.
3. Na janela Implantar, escolha um cenário.

4. Insira um Nome de Jogo. Essa opção deve ser exclusiva. O nome do jogo faz parte do nome da pilha AWS CloudFormation quando você implanta o cenário.
5. Escolha o Caminho da pasta de compilação do servidor de jogo. O caminho da pasta de compilação aponta para a pasta que contém o executável e as dependências do servidor.
6. Escolha o Caminho do arquivo .exe de compilação do servidor de jogo. O caminho do arquivo executável de compilação aponta para o executável do servidor do jogo.
7. Escolha Iniciar implantação para começar a implantar um cenário. Você poderá acompanhar o status da atualização na janela Implantação em Estado atual. Os cenários podem levar até 30 minutos para serem implantados.



8. Quando a implantação do cenário é concluída, o Estado atual é atualizado para incluir o ID do cliente Cognito e a Endpoint do API Gateway, que você pode copiar e colar no jogo.



9. Para atualizar as configurações do jogo, no menu Unity, escolha Ir para configurações de conexão do cliente. Isso exibe um guia Inspector no lado direito da tela do Unity.
10. Desmarque o Modo de teste local.
11. Insira o Endpoint do API Gateway e o ID do cliente Cognito. Escolha a mesma Região da AWS que você usou para a implantação do cenário. Em seguida, você poderá reconstruir e executar o cliente do jogo usando os recursos do cenário implantado.

Excluindo recursos criados pelo cenário

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Para excluir os recursos criados para o cenário, exclua a pilha AWS CloudFormation correspondente.

Para excluir recursos criados pelo cenário

1. Na janela Amazon GameLift Plugin for Unity Deployment, selecione View AWS CloudFormation Console para abrir o AWS CloudFormation console.
2. No console do AWS CloudFormation, escolha Pilhas e, em seguida, escolha a pilha que inclui o nome do jogo especificado durante a implantação.
3. Escolha Excluir para excluir a pilha. Pode levar alguns minutos para excluir uma pilha. Depois o AWS CloudFormation excluir a pilha usada pelo cenário, seu status muda para ROLLBACK_COMPLETE.

Integre jogos com a Amazon GameLift no Unity

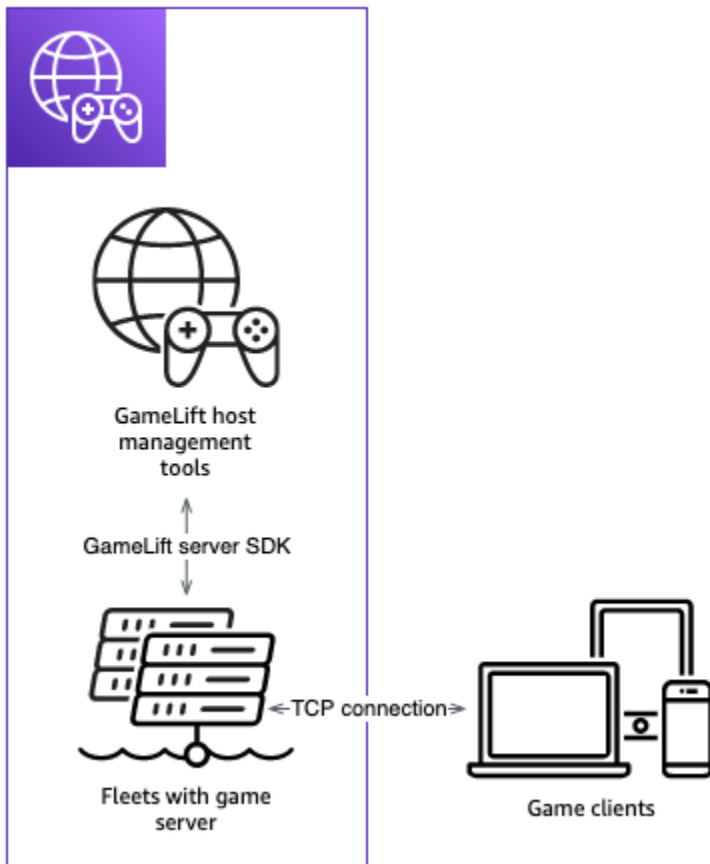
Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Integre seu jogo Unity com a Amazon GameLift concluindo as seguintes tarefas:

- [Integre a Amazon GameLift com um projeto de servidor de jogos Unity](#)
- [Integre a Amazon GameLift com um projeto de cliente de jogo Unity](#)

O diagrama a seguir mostra um exemplo de fluxo de integração de um jogo. No diagrama, uma frota com o servidor do jogo é implantada na Amazon GameLift. O cliente do jogo se comunica com o servidor do jogo, que se comunica com a Amazon. GameLift



Importe e execute um jogo de amostra

i Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

O GameLift plug-in da Amazon para Unity inclui um jogo de amostra que você pode usar para explorar os fundamentos da integração do seu jogo com a Amazon. GameLift Nesta seção, você cria o cliente e o servidor do jogo e depois testa localmente usando o Amazon GameLift Local.

Pré-requisitos

- [Configurar um Conta da AWS](#)
- [Instale e configure o plug-in](#)

Criar e executar a aplicação de exemplo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Configure os arquivos do servidor do jogo de amostra.

1. No Unity, no menu, escolha Amazon e GameLift, em seguida, escolha Importar jogo de amostra.
2. Na janela Importar amostra de jogo, escolha Importar para importar o jogo, seus ativos e dependências.
3. Crie o servidor do jogo. No Unity, no menu, escolha Amazon GameLift e escolha Aplicar configurações de compilação do Windows Sample Server ou Aplicar configurações de compilação do macOS Sample Server. Depois de definir as configurações do servidor do jogo, o Unity recompila os ativos.
4. No Unity, no menu, escolha Arquivo e, em seguida, escolha Compilar. Escolha Compilação de servidor, escolha Compilar e, em seguida, escolha uma pasta de compilação específica para arquivos do servidor.

O Unity cria o servidor de jogo de amostra, colocando os ativos executáveis e necessários na pasta de compilação especificada.

Criar e executar a aplicação do jogo de exemplo

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Configure os arquivos do cliente do jogo de amostra.

1. No Unity, no menu, escolha Amazon e, em seguida GameLift, escolha Aplicar configurações de compilação de cliente de amostra do Windows ou Aplicar configurações de compilação de cliente de amostra do macOS. Depois que as configurações do cliente do jogo forem definidas, o Unity recompilará os ativos.

2. No Unity, no menu, selecione Ir para configurações do cliente. Isso exibirá um guia Inspector no lado direito da tela do Unity. Na guia Amazon GameLift Client Settings, selecione Modo de teste local.
3. Crie o cliente do jogo. No Unity, no menu, escolha Arquivo. Confirme se a opção Compilação de servidor não está marcada, escolha Compilação e selecione uma pasta de compilação específica para arquivos do cliente.

O Unity cria o cliente de jogo de amostra, colocando os ativos executáveis e necessários na pasta de criação do cliente especificada.

4. Você não construiu o servidor e o cliente do jogo. Nas próximas etapas, você executa o jogo e vê como ele interage com a Amazon GameLift.

Teste o jogo de amostra localmente

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Execute o jogo de amostra que você importou usando o Amazon GameLift Local.

1. Inicie o servidor do jogo. No Unity, no guia Plug-in para Unity, escolha o guia Implantar.
2. No painel Teste, escolha Abrir interface de teste local.
3. Na janela Teste local, especifique um Caminho de arquivo .exe do servidor de jogos. O caminho deve incluir o nome do executável. Por exemplo, `C:/MyGame/GameServer/MyGameServer.exe`.
4. Escolha Implantar e executar. O plug-in para Unity inicia o servidor do jogo e abre uma janela de registro GameLift local da Amazon. A janela contém mensagens de log, incluindo mensagens enviadas entre o servidor do jogo e o Amazon GameLift Local.
5. Inicie o cliente do jogo. Encontre o local de compilação com o cliente do jogo de amostra e escolha o arquivo executável.
6. No Amazon GameLift Sample Game, forneça um e-mail e uma senha e escolha Login. O e-mail e a senha não são validados nem usados.

7. No Amazon GameLift Sample Game, escolha Start. O cliente do jogo procura uma sessão de jogo. Se ele não conseguir encontrar uma sessão, ele cria uma. O cliente do jogo então inicia a sessão do jogo. Você poderá ver a atividade do jogo nos registros.

Exemplos de registros do servidor de jogos

```
...
2021-09-15T19:55:3495 PID:20728 Log :) GAMELIFT AWAKE
2021-09-15T19:55:3512 PID:20728 Log :) I AM SERVER
2021-09-15T19:55:3514 PID:20728 Log :) GAMELIFT StartServer at port 33430.
2021-09-15T19:55:3514 PID:20728 Log :) SDK VERSION: 4.0.2
2021-09-15T19:55:3556 PID:20728 Log :) SERVER IS IN A GAMELIFT FLEET
2021-09-15T19:55:3577 PID:20728 Log :) PROCESSREADY SUCCESS.
2021-09-15T19:55:3577 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
...
2021-09-15T19:55:3634 PID:20728 Log :) GAMELOGIC AWAKE
2021-09-15T19:55:3635 PID:20728 Log :) GAMELOGIC START
2021-09-15T19:55:3636 PID:20728 Log :) LISTENING ON PORT 33430
2021-09-15T19:55:3636 PID:20728 Log SERVER: Frame: 0 HELLO WORLD!
...
2021-09-15T19:56:2464 PID:20728 Log :) GAMELIFT SESSION REQUESTED
2021-09-15T19:56:2468 PID:20728 Log :) GAME SESSION ACTIVATED
2021-09-15T19:56:3578 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
2021-09-15T19:57:3584 PID:20728 Log :) GAMELIFT HEALTH CHECK REQUESTED (HEALTHY)
2021-09-15T19:58:0334 PID:20728 Log SERVER: Frame: 8695 Connection accepted: playerId
 0 joined
2021-09-15T19:58:0335 PID:20728 Log SERVER: Frame: 8696 Connection accepted: playerId
 1 joined
2021-09-15T19:58:0338 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from playerId 0 Msg:
CONNECT: server IP localhost
2021-09-15T19:58:0338 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from player 0:CONNECT:
server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 CONNECT: player index 0
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from playerId 1 Msg:
CONNECT: server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 Msg rcvd from player 1:CONNECT:
server IP localhost
2021-09-15T19:58:0339 PID:20728 Log SERVER: Frame: 8697 CONNECT: player index 1
```

Exemplos de registros GameLift locais da Amazon

```
12:55:26,000 INFO || - [SocketIOServer] main - Session store / pubsub factory used:
MemoryStoreFactory (local session store only)
12:55:28,092 WARN || - [ServerBootstrap] main - Unknown channel option 'SO_LINGER' for
channel '[id: 0xe23d0a14]'
12:55:28,101 INFO || - [SocketIOServer] nioEventLoopGroup-2-1 - SocketIO server
started at port: 5757
12:55:28,101 INFO || - [SDKConnection] main - GameLift SDK server (communicates with
your game server) has started on http://localhost:5757
12:55:28,120 INFO || - [SdkWebSocketServer] WebSocketSelector-20 - WebSocket Server
started on address localhost/127.0.0.1:5759
12:55:28,166 INFO || - [StandAloneServer] main - GameLift Client server (listens for
GameLift client APIs) has started on http://localhost:8080
12:55:28,179 INFO || - [StandAloneServer] main - GameLift server sdk http listener has
started on http://localhost:5758
12:55:35,453 INFO || - [SdkWebSocketServer] WebSocketWorker-12 - onOpen
socket: /?pID=20728&sdkVersion=4.0.2&sdkLanguage=CSharp and handshake /?
pID=20728&sdkVersion=4.0.2&sdkLanguage=CSharp
12:55:35,551 INFO || - [HostProcessManager] WebSocketWorker-12 - client connected with
pID 20728
12:55:35,718 INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
GameLift API to use: ProcessReady for pId 20728
12:55:35,718 INFO || - [ProcessReadyHandler] GameLiftSdkHttpHandler-thread-0 -
Received API call for processReady from 20728
12:55:35,738 INFO || - [ProcessReadyHandler] GameLiftSdkHttpHandler-thread-0 -
onProcessReady data: port: 33430
12:55:35,739 INFO || - [HostProcessManager] GameLiftSdkHttpHandler-thread-0 -
Registered new process with pId 20728
12:55:35,789 INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
GameLift API to use: ReportHealth for pId 20728
12:55:35,790 INFO || - [ReportHealthHandler] GameLiftSdkHttpHandler-thread-0 -
Received API call for ReportHealth from 20728
12:55:35,794 INFO || - [ReportHealthHandler] GameLiftSdkHttpHandler-thread-0 -
ReportHealth data: healthStatus: true
12:56:24,098 INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
GameLift.DescribeGameSessions
12:56:24,119 INFO || - [DescribeGameSessionsDispatcher] Thread-12 - Received API call
to describe game sessions with input: {"FleetId":"fleet-123"}
12:56:24,241 INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
GameLift.CreateGameSession
12:56:24,242 INFO || - [CreateGameSessionDispatcher] Thread-12 - Received API call to
create game session with input: {"FleetId":"fleet-123","MaximumPlayerSessionCount":4}
```

```
12:56:24,265 INFO || - [HostProcessManager] Thread-12 - Reserved process:
 20728 for gameSession: arn:aws:gamelift:local::gamesession/fleet-123/
gssess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d
12:56:24,266 INFO || - [WebSocketInvoker] Thread-12 - StartGameSessionRequest:
 gameSessionId=arn:aws:gamelift:local::gamesession/fleet-123/
gssess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d, fleetId=fleet-123, gameSessionName=null,
 maxPlayers=4, properties=[], ipAddress=127.0.0.1, port=33430, gameSessionData?=false,
 matchmakerData?=false, dnsName=localhost
12:56:24,564 INFO || - [CreateGameSessionDispatcher] Thread-12 - GameSession with
 id: arn:aws:gamelift:local::gamesession/fleet-123/gssess-59f6cc44-4361-42f5-95b5-
fdb5825c0f3d created
12:56:24,585 INFO || - [GameLiftHttpHandler] Thread-12 - API to use:
 GameLift.DescribeGameSessions
12:56:24,585 INFO || - [DescribeGameSessionsDispatcher] Thread-12 - Received API call
 to describe game sessions with input: {"FleetId":"fleet-123"}
12:56:24,660 INFO || - [GameLiftSdkHttpHandler] GameLiftSdkHttpHandler-thread-0 -
 GameLift API to use: GameSessionActivate for pId 20728
12:56:24,661 INFO || - [GameSessionActivateHandler] GameLiftSdkHttpHandler-thread-0 -
 Received API call for GameSessionActivate from 20728
12:56:24,678 INFO || - [GameSessionActivateHandler] GameLiftSdkHttpHandler-thread-0
 - GameSessionActivate data: gameSessionId: "arn:aws:gamelift:local::gamesession/
fleet-123/gssess-59f6cc44-4361-42f5-95b5-fdb5825c0f3d"
```

Desligar um processo do servidor

Note

Este tópico se refere ao GameLift plug-in Amazon para Unity versão 1.0.0, que usa o SDK do servidor 4.x ou anterior.

Depois de terminar seu jogo de amostra, desligue o servidor no Unity.

1. No cliente do jogo, escolha Sair ou feche a janela para parar o cliente do jogo.
2. No Unity, na janela Teste local, escolha Parar ou feche as janelas do servidor de jogos para parar o servidor.

Integrando jogos com o GameLift plugin Amazon para Unreal Engine

Os tópicos desta seção descrevem o GameLift plug-in da Amazon para Unreal Engine (UE) e como usá-lo para preparar seu projeto de jogo multijogador para hospedagem na Amazon GameLift. Trabalhe inteiramente em seu ambiente de desenvolvimento de UE com os fluxos de trabalho guiados do plug-in para concluir os requisitos básicos de hospedagem na Amazon GameLift.

GameLift A Amazon é um serviço totalmente gerenciado que permite aos desenvolvedores de jogos gerenciar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões. Para obter mais informações sobre GameLift hospedagem na Amazon, consulte [Como o Amazon GameLift funciona](#).

Tópicos

- [Sobre o plug-in](#)
- [Fluxo do plug-in](#)
- [Instale o plug-in para Unreal](#)
- [Configurar um perfil de usuário da AWS](#)
- [Configure seu jogo para testes com a Amazon GameLift Anywhere](#)
- [Implante seu jogo em hospedagem em nuvem com frotas de EC2 gerenciadas](#)

Sobre o plug-in

O plug-in adiciona GameLift ferramentas e funcionalidades da Amazon ao editor UE. Os fluxos de trabalho guiados do plug-in para integrar a Amazon GameLift ao seu projeto de jogo, designar uma estação de trabalho como hospedeira local para testes e implantar o servidor do jogo na hospedagem em nuvem da Amazon GameLift .

Use as soluções de hospedagem pré-criadas do plug-in para implantar seu jogo. Configure uma frota do Amazon GameLift Anywhere com sua estação de trabalho local como anfitriã. Para hospedagem na nuvem, escolha entre dois cenários de implantação comuns que equilibram a latência do jogador, a disponibilidade da sessão de jogo e o custo de maneiras diferentes. Um cenário inclui um simples FlexMatch matchmaker e um conjunto de regras. Use essas soluções para começar rapidamente com uma estrutura de hospedagem pronta para produção e, em seguida, otimize e personalize conforme necessário.

O plug-in inclui os seguintes componentes:

- Módulos de plug-in para o editor do UE. Quando o plug-in é instalado, um novo botão do menu principal dá acesso à GameLift funcionalidade da Amazon.
- Bibliotecas C++ para a API de GameLift serviços da Amazon com funcionalidade do lado do cliente.
- Bibliotecas irrealis para o SDK GameLift do servidor Amazon (versão 5).
- Conteúdo para teste, incluindo um mapa do jogo de startup e dois mapas de teste com esquemas básicos e elementos de interface do usuário para uso no teste de uma integração de servidor.
- Configurações editáveis, na forma de modelos AWS CloudFormation, que o plug-in usa ao implantar seu servidor de jogos para hospedagem.

Fluxo do plug-in

As etapas a seguir descrevem uma abordagem típica para integrar e implantar um projeto de jogo com o GameLift plug-in da Amazon para Unreal Engine. Você conclui essas etapas trabalhando no editor UE e no código do jogo.

1. Crie um perfil de usuário vinculado à sua AWS conta e forneça credenciais de acesso para um usuário de conta válido com permissões para usar a Amazon GameLift.
2. Adicione o código do servidor ao seu projeto de jogo para estabelecer comunicação entre um servidor de jogo em execução e o GameLift serviço com a Amazon.
3. Adicione o código do cliente ao seu projeto de jogo para permitir que os clientes do jogo enviem solicitações GameLift à Amazon para iniciar novas sessões de jogo e depois se conectar a elas.
4. Use o fluxo de trabalho Anywhere para configurar sua estação de trabalho local como host Anywhere para seu servidor de jogos. Inicie seu servidor e cliente de jogos localmente por meio do plug-in, conecte-se a uma sessão de jogo e teste sua integração.
5. Use o fluxo de trabalho de hospedagem EC2 para carregar seu servidor de jogos integrado e implantar uma solução de hospedagem em nuvem. Quando o servidor de jogos estiver pronto, inicie seu cliente de jogo localmente por meio do plug-in, conecte-se a uma sessão de jogo e jogue o jogo.

Ao trabalhar no plug-in, você criará e usará AWS recursos. Essas ações podem incorrer em cobranças na AWS conta em uso. Se você é novatoAWS, essas ações podem estar cobertas pelo [nível AWS gratuito](#).

Instale o plug-in para Unreal

Esta seção descreve as tarefas iniciais de instalação para adicionar o plug-in a um projeto do Unreal Engine. A funcionalidade do plug-in está disponível quando você tem o projeto aberto no editor Unreal.

Note

Você pode usar o GameLift plug-in da Amazon com uma versão padrão do editor UE, mas precisa usar uma versão original ao empacotar a compilação do servidor de jogos.

Antes de começar

Veja o que você precisa para usar o GameLift plug-in da Amazon para o Unreal Engine:

- GameLift Plugin da Amazon para o pacote de lançamento do Unreal Engine. [\[Site de download\]](#).
- Microsoft Visual Studio 2019 ou mais recente.
- Uma versão original do editor do Unreal Engine. Você precisa de uma versão original para empacotar os componentes do servidor para um jogo multijogador. Para obter mais detalhes, incluindo pré-requisitos adicionais, consulte a documentação do Unreal Engine:
 - [Acessando o código-fonte do Unreal Engine nas](#) contas GitHub Você precisará GitHub e da Epic Games.
 - Tutorial de [Compilação do Unreal Engine a partir do código-fonte](#).
- Um projeto de jogo multijogador com código de jogo C ++. Se você estiver trabalhando com um projeto Blueprint, consulte a documentação da Unreal sobre como gerar código-fonte C++ para seu projeto.

Adicione o plug-in ao projeto de jogo

Conclua as tarefas a seguir para adicionar o plug-in ao seu projeto de jogo.

Crie o SDK do servidor Amazon GameLift C++

1. Descompacte o GameLift plug-in da Amazon para o pacote de lançamento do Unreal Engine para extrair dois arquivos zip:
 - `amazon-gamelift-plugin-unreal-<>-sdk-<>.zip`

- GameLift-Cpp-ServerSDK-<>.zip.

Descompacte esses arquivos.

2. Abra a GameLift-Cpp-ServerSDK-<> pasta e conclua as seguintes instruções para sua plataforma: Linux ou Microsoft Windows.

Linux

1. Execute os seguintes comandos:

```
mkdir out
cd out
cmake -DBUILD_FOR_UNREAL=1 ..
make
```

Esses comandos criam o `/lib/aws-cpp-sdk-gamelift-server.so` arquivo.

2. Copie `/lib/aws-cpp-sdk-gamelift-server.so` para o `amazon-gamelift-plugin-unreal/GameLiftPlugin/Source/GameLiftServer/ThirdParty/GameLiftServerSDK/Linux/x86_64-unknown-linux-gnu/` diretório.

Microsoft Windows

1. Execute os seguintes comandos:

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

Esses comandos criam os seguintes arquivos binários.

- `prefix\bin\aws-cpp-sdk-gamelift-server.dll`
 - `prefix\lib\aws-cpp-sdk-gamelift-server.lib`
2. Copie os arquivos para o `amazon-gamelift-plugin-unreal\GameLiftPlugin\Source\GameLiftServer\ThirdParty\GameLiftServerSDK\Win64\` diretório.

Conclua as tarefas a seguir trabalhando nos arquivos de projeto do jogo.

1. Instale os arquivos do plug-in.
 - a. Localize a pasta raiz do seu projeto de jogo, como `... > Unreal Projects/[project-name]/`. Se a pasta `Plugins` não existir lá, crie-a.
 - b. Vá para a `amazon-gamelift-plugin-unreal` pasta descompactada. `amazon-gamelift-plugin-unreal-<>-sdk-<>.zip` Copie a `GameLiftPlugin` pasta da `gamelift-plugin-unreal` pasta para a `Plugins` pasta no diretório do projeto do jogo.
2. Adicione o plugin ao arquivo file **.uproject**.
 - a. Na pasta raiz do seu projeto de jogo, abra o arquivo `.uproject`.
 - b. Atualize o arquivo para adicionar `GameLiftPlugin` e `WebBrowserWidget` à `Plugins` seção e habilítá-los. O código a seguir mostra o `.uproject` arquivo atualizado de um jogo chamado `"MyGame"`.


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

3. Altere a versão do editor do UE para seu projeto.

Se você criou um projeto para uma versão do editor e agora quer mudar para outra versão (como uma versão de compilação de código-fonte), você precisa atualizar o projeto.

Na pasta raiz do seu projeto de jogo, selecione o arquivo `.uproject` e escolha a opção Alternar versão do Unreal Engine. Selecione uma nova versão do editor.

4. Recompile a solução do projeto com suas atualizações.
 - a. Na pasta raiz do projeto, procure um arquivo de solução (`*.sln`). Se nenhum existir, selecione o arquivo `.uproject` e escolha a opção Gerar arquivos de projeto do Visual Studio.
 - b. Abra o arquivo da solução e crie ou reconstrua o projeto.
5. Verifique se o plug-in está habilitado no editor do UE.

 Note

Se você já tiver o editor aberto, talvez seja necessário reiniciá-lo antes que ele reconheça o novo plug-in.

- a. Abra o projeto no editor do UE escolhido.
- b. Verifique na barra de ferramentas do editor principal o novo botão de GameLift menu da Amazon [imagem necessária].
- c. Procure no Navegador de conteúdo os ativos do GameLift plug-in da Amazon. Certifique-se de que sua configuração de Opções de visualização tenha a opção Mostrar conteúdo do plug-in selecionada.

Configurar um perfil de usuário da AWS

Depois de instalar o plug-in, configure um perfil e vincule-o a uma conta da AWS de usuário válida. É possível manter vários perfis, mas só pode ter um perfil ativo de cada vez. Sempre que você trabalha no plug-in, selecione um perfil para usar.

A manutenção de vários perfis permite alternar entre diferentes cenários de hospedagem. Por exemplo, você pode configurar perfis com as mesmas credenciais da AWS, mas com Regiões da AWS diferentes. Ou você poderá configurar perfis com contas da AWS diferentes ou com diferentes conjuntos de usuários/permisões.

Note

Se você instalou a AWS CLI em sua estação de trabalho e já tem um perfil configurado, o GameLift plug-in da Amazon pode detectá-la e a listará como um perfil existente. O plug-in seleciona automaticamente qualquer perfil nomeado [default]. Você poderá usar um perfil existente ou criar outro.

Para gerenciar seus perfis da AWS

1. Na barra de ferramentas principal do editor Unreal, escolha o GameLift menu Amazon e selecione Definir perfis de AWS usuário. Essa ação abre Configurações de projeto para o plug-in. Expanda a seção Perfis de usuário da AWS.
2. Se o plug-in não detectar um perfil existente, ele solicitará que você crie um. Você pode criar um novo perfil usando uma conta da AWS nova ou existente.

Note

Você precisa usar o Console de Gerenciamento da AWS para criar uma conta da AWS e criar ou atualizar um usuário com o conjunto de permissões adequado.

Ao configurar um perfil, você precisa das seguintes informações:

- Uma conta da AWS. Se precisar criar uma nova conta da AWS, siga as instruções para criar a conta. Consulte [Criar uma conta da AWS](#) para obter mais detalhes.
- Um AWS usuário com permissões para usar a Amazon GameLift e outros AWS serviços necessários. Consulte [Configurar um Conta da AWS](#) para obter instruções sobre como configurar um usuário AWS Identity and Access Management (IAM) com GameLift permissões da Amazon e acesso programático com credenciais de longo prazo.
- Credenciais para seu usuário da AWS. Essas credenciais consistem em um ID de chave de acesso da AWS e uma chave secreta AWS Consulte [Obter suas chaves de acesso](#) para obter mais detalhes.
- Região AWS. Essa é uma localização geográfica em que você deseja criar seus recursos de hospedagem da AWS. Durante o desenvolvimento, recomendamos usar uma região próxima à sua localização física. Confira a lista de [regiões da AWS com suporte](#).

3. Se o plug-in detectar um perfil existente, você não será solicitado a criar um. Se você quiser atualizar um perfil ou criar um, escolha Gerenciar seu perfil.

Para inicializar seu perfil:

Todos os perfis devem ser inicializados para serem usados com o plug-in da Amazon. GameLift O Bootstrapping cria um bucket do Amazon S3 específico para o perfil. Ele é usado para armazenar configurações de projetos, criar artefatos e outras dependências. Os buckets não são compartilhados entre outros perfis.

O bootstrapping envolve a criação de novos recursos da AWS e pode incorrer em custos.

1. Na barra de ferramentas principal do editor Unreal, escolha o GameLift ícone da Amazon e selecione Definir perfis de AWS usuário. Essa ação abre Configurações de projeto para o plug-in. Expanda a seção Perfis de usuário da AWS.
2. Na seção Bootstrap seu perfil, selecione um perfil na lista suspensa e verifique o status do bootstrap. Se o status indicar que não existe um bucket, escolha o botão Bootstrap e crie um perfil para criar um bucket do Amazon S3 para o perfil selecionado.
3. Aguarde até que o status do bootstrap mude para "Active". Isso pode levar alguns minutos.

Configure seu jogo para testes com a Amazon GameLift Anywhere

Nesse fluxo de trabalho, você adiciona código de jogo de cliente e servidor para a GameLift funcionalidade da Amazon e usa o plug-in para designar sua estação de trabalho local como host de servidor de jogos de teste. Depois de concluir as tarefas de integração, use o plug-in para criar seus componentes de cliente e servidor do jogo.

Para iniciar o fluxo de trabalho do Amazon GameLift Anywhere:

- Na barra de ferramentas principal do editor Unreal, escolha o GameLift menu Amazon e selecione Host with Anywhere. Essa ação abre a página do plug-in Implantar o Anywhere, que apresenta um processo de seis etapas para integrar, criar e iniciar os componentes do jogo.

Etapa 1: configurar seu perfil

Escolha o perfil que você deseja usar ao seguir esse fluxo de trabalho. O perfil selecionado afeta todas as etapas do fluxo de trabalho. Todos os recursos que você cria são associados à AWS conta

do perfil e colocados na AWS região padrão do perfil. As permissões do usuário do perfil determinam seu acesso aos AWS recursos e ações.

1. Selecione um perfil na lista suspensa de perfis disponíveis. Se você ainda não tem um perfil ou deseja criar um novo, acesse o GameLift menu Amazon e escolha Definir perfis de AWS usuário.
2. Se o status do bootstrap não for “Ativo”, escolha o perfil do Bootstrap e aguarde até que o status mude para “Ativo”.

Etapa 2: configurar o código do jogo

Nesta etapa, você faz uma série de atualizações no código do cliente e do servidor para adicionar a funcionalidade de hospedagem. Se você ainda não configurou uma versão original do editor Unreal, o plug-in fornece links para instruções e código-fonte.

Com o plugin, pode aproveitar algumas conveniências ao integrar o código do seu jogo. Você poderá fazer uma integração mínima para configurar a funcionalidade básica de hospedagem. Você também poderá fazer uma integração personalizada mais ampla. As informações nesta seção descrevem a opção mínima de integração. Use os mapas de teste incluídos no plug-in para adicionar a GameLift funcionalidade do cliente Amazon ao seu projeto de jogo. Para integração com o servidor, use a amostra de código fornecida para atualizar o modo de jogo do seu projeto.

Integre o modo de jogo do seu servidor

Adicione um código de servidor ao seu jogo que permita a comunicação entre seu servidor de jogos e o GameLift serviço da Amazon. Seu servidor de jogo deve ser capaz de responder às solicitações da Amazon GameLift, como iniciar uma nova sessão de jogo, e também relatar o status da integridade do servidor de jogos e das conexões dos jogadores.

1. No editor de código, abra o arquivo de solução (.sln) do seu projeto de jogo, normalmente encontrado na pasta raiz do projeto. Por exemplo: `GameLiftUnrealApp.sln`.
2. Com a solução aberta, localize o arquivo de cabeçalho do modo de jogo do projeto: arquivo `[project-name]GameMode.h`. Por exemplo: `GameLiftUnrealAppGameMode.h`.
3. Altere o arquivo de cabeçalho para alinhá-lo com o código de exemplo a seguir. Certifique-se de substituir "GameLiftServer" pelo nome do seu próprio projeto. Essas atualizações são específicas para o servidor do jogo, recomendamos que você faça uma cópia de backup dos arquivos originais do modo de jogo para uso com seu cliente.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
// SPDX-License-Identifier: Apache-2.0

#pragma once

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

struct FProcessParameters;

DECLARE_LOG_CATEGORY_EXTERN(GameServerLog, Log, All);

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

public:
    AGameLiftServerGameMode();

protected:
    virtual void BeginPlay() override;

private:
    void InitGameLift();

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

4. Abra o arquivo [project-name]GameMode.cpp de origem relacionado (por exemplo GameLiftUnrealAppGameMode.cpp). Altere o código para alinhá-lo com o código de exemplo a seguir. Certifique-se de substituir "GameLiftUnrealApp" pelo nome do seu próprio projeto. Essas atualizações são específicas para o servidor do jogo, recomendamos que você faça uma cópia de backup do arquivo original para uso com seu cliente.

O código de exemplo a seguir mostra como adicionar os elementos mínimos necessários para a integração do servidor com a Amazon GameLift:

- Inicialize um cliente de GameLift API da Amazon. A `InitSDK()` chamada com os parâmetros do servidor é necessária para uma frota do Amazon GameLift Anywhere. Quando você

se conecta a uma frota Anywhere, o plug-in armazena os parâmetros do servidor como argumentos do console. O código de amostra pode acessar os valores em runtime.

- Implemente as funções de retorno de chamada necessárias para responder às solicitações do GameLift serviço da AmazonOnStartGameSession, incluindoOnProcessTerminate, e.onHealthCheck
- Ligue ProcessReady() para uma porta designada para notificar o GameLift serviço da Amazon quando estiver pronto para hospedar sessões de jogo.

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

#include "GameLiftServerGameMode.h"

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

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

#include "GenericPlatform/GenericPlatformOutputDevices.h"

DEFINE_LOG_CATEGORY(GameServerLog);

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

    if (PlayerPawnBPClass.Class != NULL)
    {
        DefaultPawnClass = PlayerPawnBPClass.Class;
    }

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

void AGameLiftServerGameMode::BeginPlay()
```

```
{
    Super::BeginPlay();

#if WITH_GAMELIFT
    InitGameLift();
#endif
}

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

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

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

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

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

        // If GameLift Anywhere is enabled, parse command line arguments and pass
        them in the ServerParameters object.
        FString glAnywhereWebSocketUrl = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereWebSocketUrl="),
glAnywhereWebSocketUrl))
        {
            ServerParametersForAnywhere.m_webSocketUrl =
TCHAR_TO_UTF8(*glAnywhereWebSocketUrl);
        }

        FString glAnywhereFleetId = "";
```

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

        FString glAnywhereProcessId = "";
        if (FParse::Value(FCommandLine::Get(), TEXT("glAnywhereProcessId="),
glAnywhereProcessId))
        {
            ServerParametersForAnywhere.m_processId =
TCHAR_TO_UTF8(*glAnywhereProcessId);
        }
        else
        {
            // If no ProcessId is passed as a command line argument, generate a
            randomized unique string.
            ServerParametersForAnywhere.m_processId =
                TCHAR_TO_UTF8(
                    *FText::Format(
                        FText::FromString("ProcessId_{0}"),
                        FText::AsNumber(std::time(nullptr))
                    ).ToString()
                );
        }

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

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

        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_YELLOW);
```



```

        UE_LOG(GameServerLog, Log, TEXT(">>>> Web Socket URL: %s"),
*ServerParametersForAnywhere.m_webSocketUrl);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Fleet ID: %s"),
*ServerParametersForAnywhere.m_fleetId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Process ID: %s"),
*ServerParametersForAnywhere.m_processId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Host ID (Compute Name): %s"),
*ServerParametersForAnywhere.m_hostId);
        UE_LOG(GameServerLog, Log, TEXT(">>>> Auth Token: %s"),
*ServerParametersForAnywhere.m_authToken);
        UE_LOG(GameServerLog, SetColor, TEXT("%s"), COLOR_NONE);
    }

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

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

    ProcessParameters = MakeShared<FProcessParameters>();

    //When a game session is created, GameLift sends an activation request to the
    game server and passes along the game session object containing game properties
    and other settings.
    //Here is where a game server should take action based on the game session
    object.

```

```

    //Once the game server is ready to receive incoming player connections, it
    should invoke GameLiftServerAPI.ActivateGameSession()
    ProcessParameters->OnStartGameSession.BindLambda( [=]
(Aws::GameLift::Server::Model::GameSession InGameSession)
    {
        FString GameSessionId = FString(InGameSession.GetGameSessionId());
        UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"),
*GameSessionId);
        GameLiftSdkModule->ActivateGameSession();
    });

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

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

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

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

    for (FString SwitchStr : CommandLineSwitches)

```

```
{
    FString Key;
    FString Value;

    if (SwitchStr.Split("=", &Key, &Value))
    {
        if (Key.Equals("port"))
        {
            ProcessParameters->port = FString::Atoi(*Value);
        }
    }
}

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

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

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

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

```
}
```

Integre o mapa de jogo do seu cliente

O mapa do jogo de startup contém elementos lógicos e de interface do usuário do Blueprint que já incluem código básico para solicitar sessões de jogo e usar informações de conexão para se conectar a uma sessão de jogo. Você poderá usar o mapa como está ou modificá-lo conforme necessário. Use o mapa do jogo de startup com outros ativos do jogo, como o projeto modelo de terceira pessoa fornecido pelo Unreal Engine. Esses ativos estão disponíveis no Content Browser. Você pode usá-los para testar os fluxos de trabalho de implantação do plug-in ou como um guia para criar um serviço de back-end personalizado para seu jogo.

O mapa de startup de arquivo tem as seguintes características:

- Inclui lógica para uma frota Anywhere e uma frota EC2 gerenciada. Ao administrar seu cliente, você pode optar por se conectar a qualquer frota.
- A funcionalidade do cliente inclui encontrar uma sessão de jogo `SearchGameSessions (())`, criar uma nova sessão de jogo `CreateGameSession (())` e participar diretamente de uma sessão de jogo.
- Ele obtém um ID de jogador exclusivo do grupo de usuários do Amazon Cognito do seu projeto (isso faz parte de uma solução implantada em qualquer lugar).

Para usar o mapa do jogo de startup

1. No editor do UE, abra a página Configurações do projeto, mapas e modos e expanda a seção Mapas padrão.
2. Para Editor Startup Map, selecione `StartupMap ""` na lista suspensa. Talvez seja necessário pesquisar o arquivo, que está localizado em `... > Unreal Projects/[project-name]/Plugins/Amazon GameLift Plugin Content/Maps`.
3. Para Mapa padrão do jogo, selecione o mesmo "StartupMap" na lista suspensa.
4. Para Mapa padrão do servidor, selecione "ThirdPersonMap". Esse é um mapa padrão incluído no seu projeto de jogo. Este mapa foi projetado para dois jogadores no jogo.
5. Abra o painel de detalhes do mapa padrão do servidor. Defina `GameMode Override` como "Nenhum".

6. Expanda a seção Modos padrão e defina o Modo de jogo padrão global do servidor para o modo de jogo que você atualizou para a integração do servidor.

Depois de fazer essas alterações em seu projeto, você estará pronto para criar os componentes do jogo.

Compile seus componentes de jogo

1. Crie novos arquivos de destino do servidor e do cliente
 - a. Na pasta do projeto do jogo, vá até a pasta Origem e encontre os arquivos `Target.cs`.
 - b. Copie o arquivo `[project-name]Editor.Target.cs` em dois novos arquivos chamados `[project-name]Client.Target.cs` e `[project-name]Server.Target.cs`.
 - c. Edite cada um dos novos arquivos para atualizar os valores do nome da classe e do tipo de destino, conforme mostrado:

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

using UnrealBuildTool;
using System.Collections.Generic;

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

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

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

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

2. Atualize o arquivo .Build.cs.

- a. Abra o arquivo .Build.cs de seu projeto. Esse arquivo está localizado em UnrealProjects/[project name]/Source/[project name]/[project name].Build.cs.
- b. Atualize a classe ModuleRules conforme mostrado no exemplo de código a seguir.

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

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

3. Crie a solução de projeto de jogo novamente.
4. Abra seu projeto de jogo em uma versão original do editor do Unreal Engine.
5. Faça o seguinte para seu cliente e servidor:

- a. Escolha um destino. Vá para Plataformas, Windows e selecione uma das seguintes opções:
 - Servidor: `[your-application-name]Server`
 - Cliente: `[your-application-name]Client`
- b. Inicie a compilação. Vá para Plataforma, Windows, Projeto de Pacote.

Cada processo de empacotamento gera um executável: `[your-application-name]Client.exe` ou `[your-application-name]Server.exe`.

No plug-in, defina os caminhos para os executáveis de criação do cliente e do servidor em sua estação de trabalho local.

Etapa 3: conectar-se a uma frota do Anywhere

Nesta etapa, você designa uma frota do Anywhere para usar. Uma frota do Anywhere define uma coleção de recursos computacionais, que podem estar localizados em qualquer lugar, para hospedagem de servidores de jogos.

- Se a AWS conta que você está usando atualmente tiver frotas Anywhere existentes, abra o campo suspenso Nome da frota e escolha uma frota. Essa lista suspensa mostra apenas as frotas do Anywhere na região da AWS para o perfil de usuário atualmente ativo.
- Se não houver frotas existentes, ou se você quiser criar uma, escolha Criar frota de Anywhere e forneça um nome para a frota.

Depois de escolher uma frota Anywhere para seu projeto, a Amazon GameLift verifica se o status da frota está ativo e exibe o ID da frota. Você pode acompanhar o progresso dessa solicitação no registro de saída do editor Unreal.

Etapa 4: registrar sua estação de trabalho

Nesta etapa, você registra sua estação de trabalho local como um recurso computacional na nova frota do Anywhere.

1. Insira um nome de computação para a sua máquina local. Se você adicionar mais de uma computação na frota, os nomes deverão ser exclusivos.

2. Forneça um endereço IP para sua máquina local. Esse campo é padronizado para o endereço IP público da sua máquina. Você também pode usar o localhost (127.0.0.1), desde que esteja executando o cliente e o servidor do jogo na mesma máquina.
3. Escolha Registrar computação. Você pode acompanhar o progresso dessa solicitação no registro de saída do editor Unreal.

Em resposta a essa ação, a Amazon GameLift verifica se pode se conectar à computação e retorna informações sobre a computação recém-registrada. Ele também cria os argumentos do console que os executáveis do seu jogo precisam ao inicializar a comunicação com o serviço da Amazon GameLift .

Etapa 5: gerar token de autenticação

Os processos do servidor de jogos que estão sendo executados em sua computação Anywhere precisam de um token de autenticação para fazer chamadas para o GameLift serviço. O plug-in gera e armazena automaticamente um token de autenticação para a frota de Anywhere sempre que você inicia o servidor do jogo a partir do plug-in. O valor do token de autenticação é armazenado como um argumento de linha de comando, que o código do servidor pode recuperar em runtime.

Você não precisa realizar nenhuma ação nesta etapa.

Etapa 6: iniciar o jogo

Neste ponto, você concluiu todas as tarefas necessárias para iniciar e jogar seu jogo multijogador em uma estação de trabalho local usando a Amazon GameLift.

1. Inicie seu servidor de jogos. O servidor do jogo notificará a Amazon GameLift quando ela estiver pronta para hospedar sessões de jogo.
2. Inicie seu cliente de jogo e use a nova funcionalidade para iniciar uma nova sessão de jogo. Essa solicitação é enviada à Amazon GameLift por meio do novo serviço de back-end. Em resposta, a Amazon GameLift liga para o servidor do jogo, executado em sua máquina local, para iniciar uma nova sessão de jogo. Quando a sessão de jogo estiver pronta para aceitar jogadores, a Amazon GameLift fornecerá informações de conexão para que o cliente do jogo participe da sessão de jogo.

Implante seu jogo em hospedagem em nuvem com frotas de EC2 gerenciadas

Nesse fluxo de trabalho, você usa o plug-in para modificar seu jogo para hospedagem em recursos computacionais baseados em nuvem gerenciados pela Amazon. GameLift Você adiciona o código de jogo do cliente e do servidor para a GameLift funcionalidade da Amazon e, em seguida, carrega sua compilação de servidor para o GameLift serviço da Amazon para implantação nos recursos baseados na nuvem. Quando esse fluxo de trabalho estiver concluído, você terá um cliente de jogo funcional que poderá se conectar aos seus servidores de jogos na nuvem.

Para iniciar o fluxo de trabalho do Amazon EC2 GameLift gerenciado pela Amazon:

- Na barra de ferramentas principal do editor Unreal, escolha o GameLift menu Amazon e selecione Host with Managed EC2. Essa ação abre a página do plug-in Implantar o Amazon EC2 Fleet, que apresenta um processo de seis etapas para integrar, criar, implantar e lançar seus componentes do jogo.

Etapa 1: configurar seu perfil

Escolha o perfil que você deseja usar ao seguir esse fluxo de trabalho. O perfil selecionado afeta todas as etapas do fluxo de trabalho. Todos os recursos que você cria são associados à AWS conta do perfil e colocados na AWS região padrão do perfil. As permissões do usuário do perfil determinam seu acesso aos AWS recursos e ações.

1. Selecione um perfil na lista suspensa de perfis disponíveis. Se você ainda não tem um perfil ou deseja criar um novo, acesse o GameLift menu Amazon e escolha Definir perfis de AWS usuário.
2. Se o status do bootstrap não for “Ativo”, escolha o perfil do Bootstrap e aguarde até que o status mude para “Ativo”.

Etapa 2: configurar o código do jogo

Nesta etapa, você faz uma série de atualizações no código do cliente e do servidor para adicionar a funcionalidade de hospedagem. Se você ainda não configurou uma versão original do editor Unreal, o plug-in fornece links para instruções e código-fonte.

Se você integrar seu jogo para uso com uma frota de Anywhere, você não precisa fazer nenhuma alteração no código do jogo. Se você estiver usando o mapa do jogo de startup, isso também funciona com implantações do EC2.

- [Configure o código do seu jogo \(Anywhere\)](#)
- [Compile seus componentes de jogo](#)

Depois de criar seu servidor de jogo, conclua as tarefas a seguir para prepará-lo para ser carregado na Amazon GameLift.

Para empacotar sua compilação de servidor para implantação na nuvem

Na pasta `WindowsServer` em que o editor do Unreal empacota os arquivos de compilação do seu servidor por padrão, faça as seguintes adições

1. Copie o script de instalação, incluído no download do plug-in, na raiz da pasta `WindowsServer`. Procure o arquivo `[project-name]/Plugins/Resources/CloudFormation/extra_server_resources/install.bat`. A Amazon GameLift usa esse arquivo para instalar o servidor criado em cada recurso de hospedagem EC2.
2. Copie o arquivo `VC_redist.x64.exe`, incluído na instalação do Visual Studio, na raiz da pasta `WindowsServer`. Esse arquivo geralmente está localizado em `C:/Program Files (x86)/Microsoft Visual Studio/2019/Professional/VC/Redist/MSVC/v142`.
3. Copie as DLLs do OpenSSL da sua compilação de servidor de jogos na pasta `WindowsServer/MyGame/Binaries/Win64`. Certifique-se de que as DLLs sejam da mesma versão usada na compilação do servidor. Copie os arquivos a seguir:
 - `libssl-3-x64.dll`
 - `libcrypto-3-x64.dll`

Etapa 3: selecionar o cenário de implantação

Nesta etapa, você escolhe a solução de hospedagem de jogos que deseja implantar no momento. Você poderá ter várias implantações do seu jogo, usando qualquer um dos cenários.

- Frota de região única: implanta seu servidor de jogo em uma única frota de recursos de hospedagem na região padrão AWS do perfil ativo. Esse cenário é um bom ponto de partida para

testar a integração do servidor da AWS e a configuração de compilação do servidor. Ele implanta os seguintes recursos:

- frota (sob demanda) da AWS com a versão do seu servidor de jogos instalada e em execução.
- Grupo de usuários e cliente do Amazon Cognito para permitir que os jogadores se autentiquem e iniciem um jogo.
- Autorizador de gateway de API que vincula o grupo de usuários às APIs.
- WebACL para controle de utilização excessiva de jogadores para o gateway da API.
- Gateway de API + função do Lambda para os jogadores solicitarem um slot de jogo. Essa função é chamada `CreateGameSession()` se nenhuma estiver disponível.
- Gateway de API + função do Lambda para que os jogadores obtenham informações de conexão para sua solicitação de jogo.
- FlexMatch frota: implanta seu servidor de jogo em um conjunto de frotas e configura um FlexMatch matchmaker com regras para criar partidas de jogadores. Esse cenário usa hospedagem spot de baixo custo com uma estrutura de várias frotas e vários locais para uma disponibilidade durável. Essa abordagem é útil quando você está pronto para começar a projetar um componente matchmaker para sua solução de hospedagem. Nesse cenário, você criará os recursos básicos para essa solução, que poderão ser personalizados posteriormente, conforme necessário. Ele implanta os seguintes recursos:
 - FlexMatch configuração de matchmaking e regras de matchmaking definidas para aceitar solicitações de jogadores e formar partidas.
 - Três frotas da AWS com sua versão de servidor de jogo instalada e funcionando em vários locais. Inclui duas frotas spot e uma frota sob demanda como backup.
 - Fila de posicionamento de sessões de jogo da AWS que atende às solicitações de partidas propostas, encontrando o melhor recurso de hospedagem possível (com base na viabilidade, custo, latência do jogador etc.) e iniciando uma sessão de jogo.
 - Grupo de usuários e cliente do Amazon Cognito para permitir que os jogadores se autentiquem e iniciem um jogo.
 - Autorizador de gateway de API que vincula o grupo de usuários às APIs.
 - WebACL para controle de utilização excessiva de jogadores para o gateway da API.
 - Gateway de API + função do Lambda para os jogadores solicitarem um slot de jogo. Esta função chama `StartMatchmaking()`.
 - Gateway de API + função do Lambda para que os jogadores obtenham informações de conexão para sua solicitação de jogo.

- Tabelas do Amazon DynamoDB para armazenar ingressos de marcação de jogos para jogadores e informações da sessão do jogo.
- Tópico do SNS + Função Lambda para GameSessionQueue lidar com eventos.

Etapa 4: configurar parâmetros do jogo

Nesta etapa, você descreve seu jogo para o qual fazer o upload da AWS.

- Nome da compilação do servidor: forneça um nome significativo para a compilação do servidor do jogo. AWS usa esse nome para se referir à cópia da compilação do seu servidor que é carregada e usada para implantações.
- Sistema operacional de criação de servidor: insira o sistema operacional no qual o servidor foi criado para ser executado. Isso informa a AWS que tipo de recursos computacionais usar para hospedar seu jogo.
- Pasta do servidor do jogo: identifique o caminho para a pasta de criação do servidor local.
- Compilação do servidor de jogo: identifique o caminho para o executável do servidor de jogos.
- Caminho do cliente do jogo: identifique o caminho para o executável do cliente do jogo.
- Saída de configuração do cliente: esse campo precisa apontar para uma pasta na compilação do cliente que contém sua configuração da AWS. Procure-o no seguinte local: `[client-build]/[project-name]/Content/CloudFormation`.

Etapa 5: implantar o cenário

Nesta etapa, você implanta seu jogo em uma solução de hospedagem em nuvem com base no cenário de implantação escolhido. Esse processo pode levar até 40 minutos enquanto a AWS valida a criação do servidor, provisiona recursos de hospedagem, instala o servidor do jogo, inicia os processos do servidor e os prepara para hospedar sessões de jogo.

Para iniciar a implantação, escolha Implantar CloudFormation. É possível acompanhar o status de sua hospedagem de jogos aqui. Para obter informações mais detalhadas, você pode entrar na AWS em console de gerenciamento da AWS e visualizar notificações de eventos. Certifique-se de fazer login usando a mesma conta, usuário e região da AWS, como perfil de usuário ativo no plug-in.

Quando a implantação estiver concluída, você terá seu servidor de jogos instalado em uma instância do EC2 da AWS. Pelo menos um processo do servidor está em execução e pronto para iniciar uma sessão de jogo.

Etapa 6: Inicializar o cliente

Neste ponto, você concluiu todas as tarefas necessárias para iniciar e jogar seu jogo multijogador hospedado na Amazon GameLift. Para jogar, inicie uma instância do seu cliente de jogo.

Se você implantou o cenário de frota única, poderá abrir uma única instância de cliente com um jogador, entrar no mapa do servidor e se movimentar. Abra instâncias adicionais do cliente do jogo para adicionar um segundo jogador ao mesmo mapa de jogo do servidor.

Se você implantou o FlexMatch cenário, a solução espera que pelo menos dois clientes sejam colocados na fila para a colocação da sessão de jogo antes que os jogadores possam entrar no mapa do servidor.

Obtenha dados da frota para uma instância do Amazon GameLift

Há algumas situações em que sua compilação de jogo personalizada ou script de servidores em tempo real podem exigir informações sobre a frota do Amazon GameLift. Por exemplo, a compilação ou o script do seu jogo podem incluir código para:

- Monitore a atividade com base nos dados da frota.
- Acumule métricas para rastrear a atividade por dados da frota. (Muitos jogos usam esses dados para atividades do LiveOps.)
- Forneça dados relevantes para serviços de jogos personalizados, como marcação de jogos, escalabilidade de capacidade adicional ou testes.

As informações da frota estão disponíveis como um arquivo JSON em cada instância nos seguintes locais:

- Windows: `C:\GameMetadata\gamelift-metadata.json`
- Linux: `/local/gamemetadata/gamelift-metadata.json`

O arquivo `gamelift-metadata.json` inclui os [atributos de um recurso de frota do Amazon GameLift](#).

Exemplo de arquivo JSON:

```
{
```

```
"buildArn": "arn:aws:gamelift:us-west-2:123456789012:build/build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
"buildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
"fleetArn": "arn:aws:gamelift:us-west-2:123456789012:fleet/fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
"fleetDescription": "Test fleet for Really Fun Game v0.8",
"fleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
"fleetName": "ReallyFunGameTestFleet08",
"fleetType": "ON_DEMAND",
"instanceRoleArn": "arn:aws:iam::123456789012:role/S3AccessForGameLift",
"instanceType": "c5.large",
"serverLaunchPath": "/local/game/reallyfungame.exe"
}
```

Adicionar a marcação de jogos do FlexMatch

Use o Amazon GameLift FlexMatch para adicionar a funcionalidade de combinação de jogadores aos seus jogos hospedados no Amazon GameLift. Use o FlexMatch com servidores de jogos personalizados ou servidores em tempo real.

O FlexMatch compara o serviço de marcação de jogos com o mecanismo de regras personalizável. Você cria como combinar jogadores com base nos atributos dos jogadores e nos modos de jogo que fazem sentido para o seu jogo. O FlexMatch gerencia os detalhes básicos da avaliação de jogadores que estão procurando um jogo, formando partidas com uma ou mais equipes e iniciando sessões de jogo para sediar as partidas.

Para usar o serviço completo do FlexMatch, tenha os recursos de hospedagem configurados com filas. O Amazon GameLift usa filas para localizar os melhores locais de hospedagem possíveis para jogos em várias regiões e tipos de computação. Em particular, as filas do Amazon GameLift podem usar dados de latência, quando fornecidos por clientes de jogos, para colocar sessões de jogo de forma que os jogadores tenham a menor latência possível ao jogar.

Para obter mais informações sobre o FlexMatch, incluindo ajuda detalhada com a integração de marcação de jogos nos jogos, consulte estes tópicos do [Guia do desenvolvedor do Amazon GameLift FlexMatch](#):

- [Como funciona o Amazon GameLift FlexMatch](#)
- [Etapas de integração do FlexMatch](#)

Gerenciando hospedagem com GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

GameLift A Amazon fornece um serviço completo de hospedagem em nuvem para oferecer suporte a soluções em contêineres para hospedagem de servidores de jogos. Com as frotas de GameLift contêineres da Amazon, você pode aproveitar os benefícios dos contêineres, como portabilidade, agilidade e tolerância a falhas.

Atributos principais

Os seguintes recursos estão disponíveis com as frotas de GameLift contêineres da Amazon.

- Desenvolva uma arquitetura de contêiner personalizada com contêineres leves para executar seu software de servidor de jogos nos recursos de GameLift hospedagem da Amazon.
- Inclua o Amazon GameLift Agent para gerenciar o ciclo de vida dos processos do servidor de jogos dentro de seus contêineres. O agente on-compute executa suas instruções sobre quando e como iniciar os processos do servidor e quantos manter para hospedagem de sessões de jogos.
- Personalize os recursos fornecidos pela Amazon GameLift para criar imagens de contêiner com seu aplicativo de servidor de jogos. Use o dockerfile fornecido para criar uma imagem de contêiner baseada em Linux. Armazene imagens para suas frotas de contêineres em um repositório privado do Amazon Elastic Container Registry (Amazon ECR).
- Ofereça experiências de baixa latência aos jogadores implantando recursos de frota de contêineres em Região da AWS qualquer zona local compatível com a Amazon. GameLift Crie frotas de contêineres em vários locais para simplificar o gerenciamento da frota. Consulte [Locais GameLift de hospedagem da Amazon](#).
- Teste suas soluções de hospedagem de jogos em contêineres com uma frota da Amazon GameLift Anywhere. Use Anywhere para testar localmente o desenvolvimento de sua solução, incluindo sua integração com o Amazon GameLift SDK e suas configurações de imagem de contêiner.
- Acompanhe o desempenho da hospedagem de jogos com métricas de desempenho específicas do contêiner. Monitore a integridade dos recursos da sua frota usando métricas de hardware.

- Use as ferramentas de posicionamento de sessões de GameLift jogo da Amazon, incluindo filas e FlexMatch matchmaking, para combinar os jogadores com as melhores sessões de jogo possíveis que estão sendo hospedadas em suas frotas de contêineres.
- Gerencie os recursos da frota de contêineres usando AWS CloudFormation modelos para a Amazon GameLift.

Usando frotas de contêineres durante a pré-visualização pública

O novo recurso de frotas de contêineres está atualmente em pré-visualização pública. Durante essa fase, os seguintes GameLift recursos da Amazon são suportados:

- Use frotas de contêineres para hospedar servidores de jogos criados para Linux. As frotas de contêineres usam Amazon_Linux_2023 e oferecem suporte a imagens de contêineres Linux. Não há suporte para contêineres do Windows.
- Integre projetos de servidores de jogos somente com a versão 5+ GameLift do SDK do servidor Amazon. As versões anteriores não são suportadas.
- Use qualquer um dos tipos de instância sob demanda do Amazon EC2 que a Amazon oferece suporte. GameLift No momento, não há suporte para frotas spot.

Como os contêineres funcionam na Amazon GameLift

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

As frotas de GameLift contêineres da Amazon são projetadas para oferecer flexibilidade na forma como você implanta e escala seus aplicativos em contêineres. Ele usa o Amazon Elastic Container Service (Amazon ECS) para gerenciar a implantação e execução de tarefas para suas frotas da Amazon. GameLift Este tópico descreve os elementos estruturais básicos para a execução de contêineres em uma frota GameLift gerenciada pela Amazon, ilustra arquiteturas comuns e descreve alguns conceitos fundamentais.

componentes da frota de contêineres

Frota

Uma frota de contêineres é uma coleção de instâncias do Amazon EC2, gerenciadas pela Amazon GameLift, que executam seus servidores de jogos em contêineres. Ao criar uma frota, você configura como a arquitetura do contêiner e o software do servidor de jogos são

implantados em cada instância da frota. Você pode implantar uma frota de contêineres em uma única Região da AWS ou em várias localizações geográficas. Você pode usar as ferramentas de escalabilidade GameLift manual ou automática da Amazon para escalar a capacidade de uma frota de contêineres de hospedar sessões de jogos e jogadores.

Instância

Uma instância do Amazon EC2 é o servidor virtual que fornece capacidade computacional para sua hospedagem de jogos. Com a Amazon GameLift, você pode escolher entre uma variedade de tipos de instância. Cada tipo de instância oferece uma combinação diferente de CPU, memória, armazenamento e capacidade de rede.

Quando você cria uma frota de contêineres, a Amazon GameLift implanta instâncias com base no tipo de instância que você escolher e na configuração da sua frota. Cada instância de frota implantada é idêntica e executa seu software de servidor de jogos em contêineres da mesma forma. O número de instâncias em uma frota determina o tamanho da frota e a capacidade de hospedagem de jogos.

Grupo de contêineres

A Amazon GameLift usa o conceito de um grupo de contêineres para descrever e gerenciar um conjunto de contêineres. Um grupo de contêineres é semelhante a uma “tarefa” ou “pod” de contêineres. Em cada grupo de contêineres, você pode definir como os contêineres compartilham os recursos de CPU e memória disponíveis. Você também pode configurar dependências entre contêineres e gerenciar o ciclo de vida do grupo de contêineres.

Os grupos de contêineres podem se replicar em cada instância da frota para otimizar o uso dos recursos. Você pode gerenciar a replicação definindo a estratégia de agendamento de um grupo de contêineres, da seguinte forma:

- Grupos de contêineres de réplica gerenciam os contêineres que executam seu aplicativo de servidor de jogos e software de suporte. Todas as frotas de contêineres devem definir um grupo de contêineres de réplica. Um grupo de réplicas pode ter várias cópias em cada instância da frota, dependendo dos requisitos do grupo de contêineres e dos recursos do tipo de instância em uso. Todos os contêineres do grupo de réplicas se escalam automaticamente em uma instância.
- Os grupos de contêineres do Daemon, que são opcionais, podem ser úteis para executar serviços em segundo plano ou programas utilitários, como para monitoramento. Seu software de servidor de jogos não depende diretamente dos processos em um grupo de daemons. Os grupos de contêineres de daemons não são replicados. Cada instância da frota tem no

máximo uma cópia do grupo de daemons. Isso significa que os contêineres em um grupo de daemons não se expandem em uma instância de frota junto com os contêineres em um grupo de réplicas.

Uma frota de contêineres deve ter um grupo de contêineres de réplica e, opcionalmente, pode ter um grupo de daemons.

Contêiner

O contêiner é o elemento mais básico de uma arquitetura baseada em contêiner. Ele consiste em uma imagem de contêiner com executáveis de software e arquivos dependentes. Ao definir um contêiner para usar com a Amazon GameLift, você configura como o software é executado no contêiner.

Cada grupo de contêineres em uma frota de contêineres deve ter um contêiner designado como “essencial”. Um contêiner essencial impulsiona o ciclo de vida de um grupo de contêineres. Se o contêiner essencial falhar, todo o grupo de contêineres será reiniciado.

Os tipos de contêineres incluem:

- O contêiner de réplica essencial inclui tudo o que você precisa para executar os processos do servidor de jogos e hospedar sessões de jogo para os jogadores. Ele inclui a versão do seu servidor de jogos, que é integrada ao SDK GameLift do servidor Amazon, e ao software dependente. Também inclui o Amazon GameLift Agent, que gerencia o ciclo de vida dos processos do seu servidor de jogos. O grupo de réplicas de contêineres de uma frota tem exatamente um contêiner de réplica essencial.
- Os contêineres de réplica não essenciais, também chamados de contêineres “auxiliares”, executam software para dar suporte ao seu aplicativo de servidor de jogos. O uso de um contêiner auxiliar permite que você execute e escale o software de suporte junto com seus servidores de jogos, mas gerencie como contêineres separados. Se esse tipo de contêiner falhar, somente o próprio contêiner será reiniciado; o grupo de contêineres não será afetado.
- Os contêineres daemon executam um serviço daemon para gerenciar processos em segundo plano. Um uso comum de um contêiner daemon é executar um [agente Amazon CloudWatch \(CloudWatch\)](#) para coletar métricas, registros e rastreamentos para seus contêineres. Os contêineres daemon podem ser essenciais ou não essenciais, dependendo de quando uma falha no contêiner deve resultar na reinicialização do grupo de contêineres.

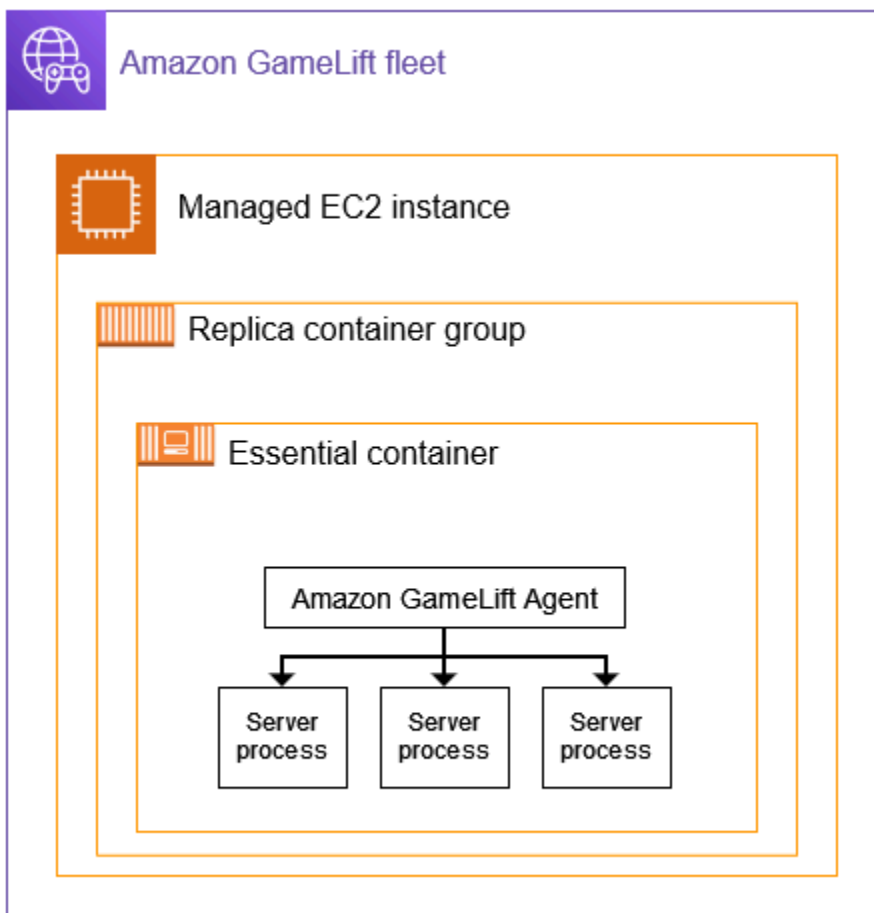
Computação

Uma computação é um recurso de hospedagem de frota registrado no GameLift serviço da Amazon e capaz de se comunicar com o serviço. Em uma frota de contêineres, uma computação

é um contêiner com um processo que gerencia o processo de registro de computação. Em um contêiner de réplica essencial de uma frota de contêineres, o Amazon GameLift Agent registra automaticamente esse contêiner como uma computação.

Arquiteturas comuns

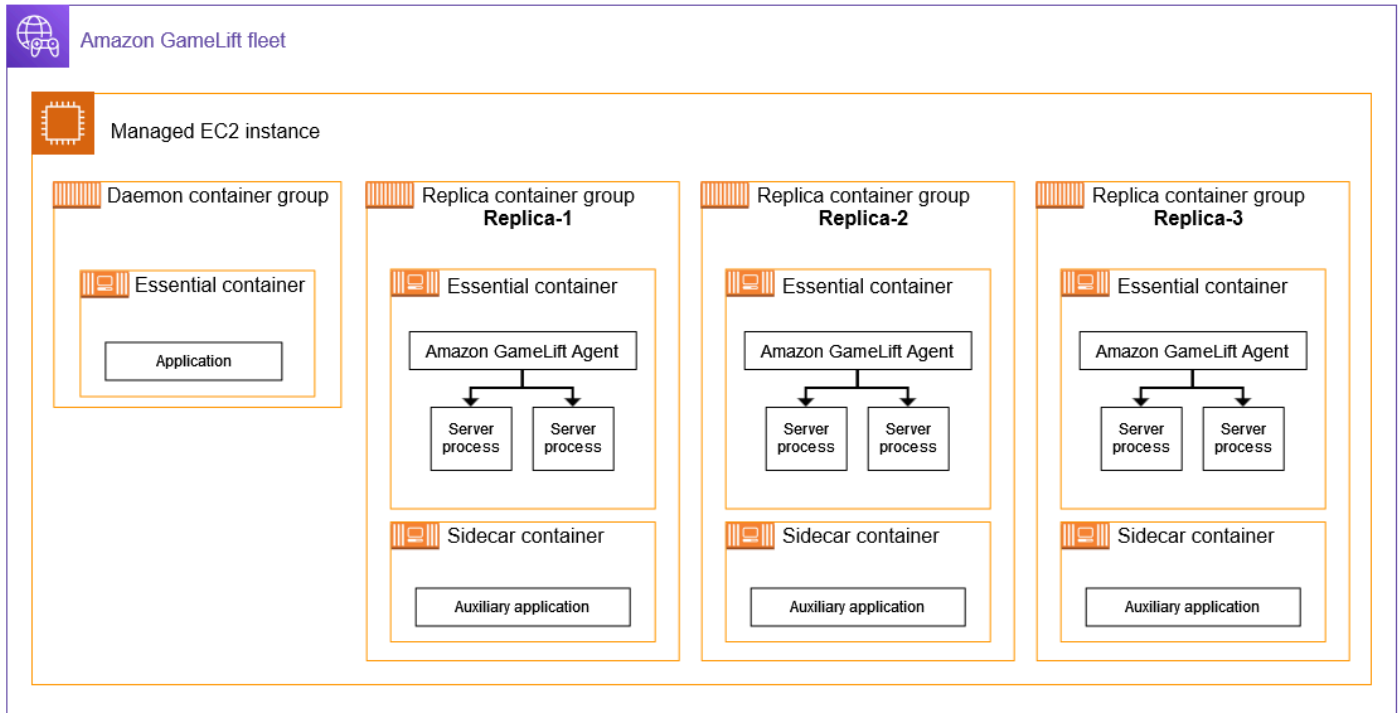
O diagrama a seguir ilustra a estrutura mais simples da frota de contêineres. Nessa estrutura, cada instância da frota mantém uma cópia do grupo de contêineres de réplica. O grupo de contêineres tem um único contêiner essencial que executa o Amazon GameLift Agent, o aplicativo de servidor de jogos e todo o software de suporte para hospedar sessões de jogos. O Agente implementa instruções específicas da frota para executar três processos do servidor simultaneamente. Como há um grupo de contêineres de réplica por instância, cada instância da frota executa três processos de servidor simultaneamente.



Este segundo exemplo ilustra um projeto de frota de contêineres mais complexo. Neste exemplo, a frota tem um grupo de contêineres de réplica com vários contêineres e um grupo de contêineres

daemon com um contêiner. A configuração da frota coloca três cópias do grupo de contêineres de réplica em cada instância da frota. O grupo de contêineres daemon nunca é replicado.

Cada conjunto de contêineres de grupos de réplicas no tem três cópias em cada instância. Em cada contêiner de réplica essencial, o agente é instruído a executar dois processos do servidor simultaneamente. Como resultado, cada instância da frota executa seis processos de servidor simultaneamente (dois processos em cada um dos três contêineres de réplica essenciais).



Conceitos principais

Esta seção resume como a Amazon GameLift implementa alguns conceitos básicos de contêineres. Para obter instruções sobre como trabalhar com frotas de contêineres, consulte os tópicos relevantes neste guia.

embalagem do grupo de contêineres

Ao desenvolver sua estrutura de contêineres para implantação em uma frota de contêineres, um objetivo comum é otimizar o uso da capacidade de computação disponível. Para atingir esse objetivo, encontre o maior número de grupos de contêineres de réplicas que você pode colocar em uma instância de frota sem afetar o desempenho do servidor de jogos.

A Amazon GameLift pode ajudar você a fazer isso. Ele calcula o número máximo de grupos de réplicas por instância, com base nas seguintes informações:

- O tipo de instância da frota e os recursos de CPU e memória disponíveis.
- Os requisitos de CPU e memória que você define para todos os contêineres em seu grupo de réplicas.

Os requisitos de CPU e memória que você define para todos os contêineres em um grupo de daemons, se houver um.

- Recursos reservados para gerenciar contêineres e outros aplicativos essenciais em cada instância.

Ao criar uma frota de contêineres, você pode optar por usar o número máximo calculado ou substituir o número calculado especificando o número desejado. Como prática recomendada, experimente seu software de servidor de jogos em contêineres para determinar os requisitos precisos de recursos. Use esses dados para encontrar uma estratégia de empacotamento ideal para o desempenho do servidor de jogos.

Servidores de jogos e o Amazon GameLift Agent

Ao criar seu contêiner de réplica essencial, você empacota seu software de servidor de jogos e o Amazon GameLift Agent juntos na mesma imagem de contêiner. Esse agente no computador controla o ciclo de vida dos servidores de jogos no contêiner. Em cada grupo de contêineres de réplica, o contêiner de réplica essencial executa o Agente e todos os processos do servidor do jogo.

O Amazon GameLift Agent executa instruções na configuração de tempo de execução da frota de contêineres. A configuração do tempo de execução identifica (1) o executável a ser executado, (2) um conjunto opcional de parâmetros de inicialização e (3) o número de processos a serem executados simultaneamente. Uma configuração de tempo de execução pode ter instruções para vários executáveis diferentes. Pelo menos uma instrução deve ser para o executável do servidor do jogo. Por exemplo, uma configuração de tempo de execução pode instruir o agente a manter 10 processos do seu servidor de jogos executáveis para uso em produção, 1 processo do mesmo executável com parâmetros de inicialização especiais para teste e 1 processo para um utilitário de registro.

Você pode modificar a configuração de tempo de execução de uma frota a qualquer momento. O GameLift agente da Amazon solicita periodicamente atualizações do serviço. Quando uma configuração de tempo de execução atualizada está disponível, o Agente a recebe e começa a implementar as instruções. As ações podem incluir adicionar ou desligar processos do servidor.

O Amazon GameLift Agent é uma versão de código aberto do agente on-compute que a Amazon GameLift usa para frotas EC2 gerenciadas. Este guia fornece instruções sobre como criar o Agente a partir do código-fonte e incorporá-lo em uma imagem de contêiner. O agente executa as seguintes tarefas:

Gerenciamento de processos do servidor:

- Inicie, desligue e substitua os processos do servidor com base na configuração do tempo de execução.
- Encerre os processos do servidor quando eles não forem ativados a tempo.
- Informe à Amazon GameLift quando um processo do servidor for encerrado.
- Emita eventos da frota para processos do servidor.

Gerenciamento de contêineres:

- Encerre os processos do servidor em resposta às solicitações da Amazon GameLift.
- Relate a integridade do contêiner.

Tarefas de upload de registros:

- Faça upload dos registros da sessão do jogo em um bucket Amazon S3 designado.
- Faça o upload dos registros do agente no computador para um bucket Amazon S3 designado.

Escalar a capacidade de frota

A capacidade da frota mede o número de sessões de jogo que a frota pode hospedar a qualquer momento. Você também pode medir a capacidade com base no número de jogadores que a frota pode suportar simultaneamente.

Para aumentar ou diminuir a capacidade de hospedagem de uma frota, você adiciona ou remove instâncias da frota. A estratégia de empacotamento de uma frota de contêineres determina quantas sessões de jogo são executadas simultaneamente em cada instância da frota. Esse número indica o número de sessões de jogo (e vagas de jogadores) que você adiciona ou subtrai ao aumentar ou diminuir a capacidade da frota.

Com frotas de contêineres, você pode usar qualquer um dos métodos de escalabilidade fornecidos pela Amazon. GameLift Isso inclui:

- Defina a capacidade da frota manualmente definindo uma contagem específica de instâncias da frota desejada.
- Configure o escalonamento automático visando o buffer desejado de instâncias disponíveis (rastreamento de metas). Esse método mantém automaticamente um conjunto de recursos de hospedagem ociosos para que os novos jogadores sempre possam entrar nos jogos rapidamente. Conforme a demanda do jogador aumenta ou diminui, o tamanho desse buffer é ajustado continuamente.
- Configure o escalonamento automático com regras de escalabilidade personalizadas (recurso avançado).

Conexões cliente/servidor do jogo

As frotas gerenciadas do EC2 e as frotas de contêineres lidam com as conexões entre clientes de jogos e servidores de jogos hospedados na nuvem de forma semelhante. Quando a Amazon GameLift cria uma nova sessão de jogo, o serviço comunica as informações de conexão da sessão de jogo. Os clientes do jogo usam as informações para se conectar diretamente ao servidor do jogo que está hospedando a sessão do jogo. Para todos os tipos de frotas, as informações de conexão consistem em um endereço IP e na atribuição de portas.

Ao criar uma frota de contêineres, você define dois conjuntos de intervalos de portas. Primeiro, você define uma variedade de portas de conexão externas que permitem que clientes de jogos se conectem a um jogo. Segundo, você define um conjunto de portas de contêiner somente internas, que são atribuídas a cada processo do servidor de jogo que está sendo executado no contêiner. A Amazon mapeia GameLift dinamicamente as portas internas do contêiner para as portas de conexão externas para dar aos jogadores acesso aos jogos. Essa abordagem fornece uma camada adicional de segurança ao proteger seus servidores de jogos do acesso direto às portas do contêiner.

Ao definir intervalos de portas para uma frota de contêineres, você deve fornecer intervalos com portas suficientes para acomodar todos os processos do servidor que são executados simultaneamente nos contêineres em uma instância.

Para controle adicional, você também define permissões de entrada para uma frota. As permissões de entrada determinam quais portas de conexão estão abertas para o tráfego de entrada. Você pode alterar as permissões de entrada de uma frota a qualquer momento. Com as permissões de entrada, você pode desligar rapidamente todas as portas de conexão, abrir algumas ou abrir todas conforme necessário.

Roteiro de desenvolvimento para contêineres da Amazon GameLift

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

O fluxo de trabalho a seguir resume as etapas para que seus servidores de jogos funcionem em uma frota de GameLift contêineres da Amazon.

Etapa 1: integre seu jogo com a Amazon GameLift

Adicione funcionalidade ao seu servidor de jogos para que ele possa se comunicar com o GameLift serviço da Amazon quando implantado em uma frota de contêineres. Se você estiver usando FlexMatch matchmaking, adicione essa funcionalidade ao seu servidor de jogo e cliente. Para obter informações detalhadas, consulte [Integre seu jogo com a Amazon GameLift](#).

- Obtenha o SDK GameLift do servidor Amazon (versão 5+) e configure-o com seu projeto de jogo. O SDK do servidor está disponível em C++, C# e Go.
- Modifique o código do servidor do jogo para adicionar a funcionalidade necessária do SDK do servidor.
- Package a versão do seu servidor de jogos para Linux. Se você estiver desenvolvendo no Windows, essa etapa pode exigir trabalho adicional para configurar um ambiente Linux.
- (Opcional) Teste a integração do seu servidor de jogos usando uma GameLift Anywhere frota da Amazon. Teste antes de preparar sua imagem de contêiner para isolar problemas com seu trabalho de integração. Para testar as conexões cliente/servidor do jogo, integre também o cliente do jogo.

Note

Se você estiver desenvolvendo no Windows, configure um espaço de trabalho Linux separado ou use uma ferramenta como o Windows Subsystem for Linux (WSL). Você precisará de um ambiente Linux para testar a compilação do seu servidor de jogos e também para criar e testar suas imagens de contêiner.

Etapa 2: Preparar a imagem do contêiner do servidor de jogos

Crie uma imagem de contêiner que execute os processos do seu servidor de jogos e armazene-a em um repositório do Amazon Elastic Container Registry (Amazon ECR) para uso com a Amazon GameLift. Para obter instruções detalhadas, consulte [Prepare uma imagem de contêiner com seu software de servidor de jogos](#).

- Configure um diretório de trabalho para sua imagem de contêiner, com a compilação do jogo Linux, o script de instalação e todos os softwares e dependências de suporte.
- Obtenha o código-fonte do Amazon GameLift Agent, crie-o e adicione o jar arquivo ao seu diretório de trabalho.
- Obtenha o Dockerfile padrão e modifique-o para configurar uma imagem de contêiner com seu software de servidor de jogos.
- Crie sua imagem de contêiner. Execute essa etapa em um ambiente Linux.
- Crie um repositório privado do Amazon ECR e envie sua imagem de contêiner para ele. Crie o repositório no mesmo Região da AWS local Conta da AWS e onde você planeja implantar sua frota de contêineres.
- (opcional) Teste as imagens do contêiner usando sua Anywhere frota. Você pode definir uma configuração de tempo de execução para passar instruções para o Amazon GameLift Agent.

Etapa 3: Crie seus contêineres e grupos de contêineres

Crie uma arquitetura de contêiner para hospedagem de jogos na Amazon GameLift. Consulte [Projete uma frota de GameLift contêineres da Amazon](#) e [Crie definições de grupos de contêineres para uma frota de GameLift contêineres da Amazon](#).

- Defina suas configurações de contêiner. Para cada contêiner, você definirá problemas como processos de tempo de execução, alocação de memória, verificações de integridade, portas de rede etc.
- Use o GameLift console da Amazon ou a AWS CLI para criar definições de grupos de contêineres com suas configurações de contêineres. Quando você cria uma definição de grupo de contêineres, a Amazon GameLift tira um instantâneo das imagens de cada contêiner naquele momento.

Etapa 4: implantar seu servidor de jogos em contêineres em uma frota de contêineres

Use as definições de grupo de contêineres criadas na etapa anterior para criar uma frota de contêineres e implantar seu software de servidor de jogos em contêineres. Consulte [Crie uma frota de GameLift contêineres da Amazon](#).

- Use o GameLift console da Amazon ou a AWS CLI para criar uma frota de contêineres.
- Acompanhe o status da frota à medida que as instâncias da frota são implantadas e ativadas. Verifique os eventos de criação da frota para verificar se a frota está sendo implantada com sucesso em todos os locais.
- Verifique se os clientes do jogo podem solicitar e participar de sessões de jogo e jogar o jogo. Se você configurou o matchmaking, teste esses cenários.

Etapa 5: gerencie suas frotas

Ao se preparar para o uso em nível de produção, desenvolva sua solução de hospedagem de jogos e gerencie seu ciclo de vida de hospedagem.

- Crie frotas em vários locais e frotas em outras Regiões da AWS para apoiar sua base de jogadores.
- Configure o posicionamento da hospedagem de jogos com filas ou FlexMatch matchmaking. Veja esses recursos:
 - [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#)
 - [FlexMatch Guia do desenvolvedor](#)
- Configure o escalonamento automático para gerenciar a capacidade da frota com base na demanda dos jogadores por sessões de jogo.
- Configure o monitoramento para suas frotas de contêineres. Trabalhe com GameLift métricas da Amazon, recupere registros de sessões de jogos e registros de contêineres, configure o acesso remoto a contêineres individuais.
- Configure o gerenciamento de longo prazo das frotas de contêineres. Use aliases de frota para agilizar o processo de atualização de frotas de contêineres. Crie AWS CloudFormation modelos para gerenciar o ciclo de vida da frota. Veja esses recursos:
 - [Adicionar um alias a uma frota da Amazon GameLift](#)
 - [Gerenciar recursos usando o AWS CloudFormation](#)

Integre seu jogo com a Amazon GameLift

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Antes de criar uma imagem de contêiner com seu software de servidor de jogos e implantá-la na Amazon GameLift para hospedagem na nuvem, primeiro integre seu projeto de jogo ao SDK do GameLift servidor Amazon e crie um servidor de jogos para ser executado no Linux. Este tópico apresenta as várias ferramentas de integração que a Amazon GameLift fornece.

Os servidores de jogos hospedados devem ser capazes de se comunicar com o GameLift serviço da Amazon. Configure a comunicação adicionando o SDK do GameLift servidor Amazon (versão 5+) ao seu projeto de jogo e modificando o código do servidor do seu jogo. GameLift A Amazon fornece recursos e documentação do SDK do servidor para oferecer suporte a várias linguagens e mecanismos de jogo.

O processo de integração para servidores de jogos em contêineres é praticamente idêntico à integração de servidores de jogos para hospedagem em frotas gerenciadas do EC2 ou da Amazon. GameLift Anywhere

Ferramentas de integração

A Amazon GameLift fornece as seguintes ferramentas e suporte linguístico para integração:

Para desenvolvedores da Unreal Engine

Use o plugin leve para Unreal. Esse plug-in inclui as bibliotecas do SDK do servidor C++ com a funcionalidade necessária da Amazon GameLift . Use a documentação para configurar seu projeto de jogo Unreal para o plug-in e atualizar o código do jogo com os blocos de código fornecidos para adicionar a funcionalidade necessária para suas compilações de servidor e cliente.

- [Download do plugin SDK](#)
- [Guia: Integre seu projeto Unreal com a Amazon GameLift](#)
- [Guia de referência: C++ Server SDK 5 para Unreal](#)

Observação: o plug-in GameLift autônomo da Amazon para Unreal Engine não suporta o uso de frotas de contêineres.

Para desenvolvedores do Unity

Use o plug-in leve para Unity. Esse plug-in inclui as bibliotecas do SDK do servidor C# com a funcionalidade necessária da Amazon GameLift . Use a documentação para configurar seu projeto de jogo Unreal para o plug-in e atualizar o código do jogo com os blocos de código fornecidos para adicionar a funcionalidade necessária para suas compilações de servidor e cliente.

- [Download do plugin SDK](#)
- [Guia: Integre seu projeto Unity com a Amazon GameLift](#)
- [Guia de referência: C# Server SDK 5 para Unity](#)

Observação: o plug-in GameLift autônomo da Amazon para Unity não suporta o uso de frotas de contêineres.

Para desenvolvedores que usam outros mecanismos de jogo

Siga esta orientação geral de integração entre servidor e cliente:

- [Integre um servidor de jogos](#)
- [Integre um cliente de jogo](#)

A Amazon GameLift oferece 5 bibliotecas de SDK de servidor para os seguintes idiomas:

- [Server SDK 5 para C++](#) [[download do SDK](#)] [[Guia de referência](#)]
- [Server SDK 5 para C#](#) [[download do SDK](#)] [[Guia de referência](#)]
- Server SDK 5 para Go [[download do SDK](#)] [[Guia de referência](#)]

Crie seu servidor de jogos para Linux

As frotas de GameLift contêineres da Amazon oferecem suporte a servidores de jogos que são executados em uma plataforma Linux. Aqui estão algumas dicas para criar seu servidor de jogos para um destino Linux:

- Se você estiver desenvolvendo seu jogo com o motor de jogo Unity, o editor de jogos fornece suporte integrado sem requisitos especiais para criar para Linux.
- Se você estiver desenvolvendo seu jogo em C++, deverá incluir as bibliotecas OpenSSL para Linux ao criar o Amazon Server GameLift SDK para C++ e ao criar seu servidor de jogos. Inclua também as mesmas bibliotecas na imagem do contêiner do seu servidor de jogos.
- Se você estiver desenvolvendo seu jogo com o Unreal Engine no Windows, considere estas opções:
 - Trabalhe com o Unreal Engine para configurar uma cadeia de ferramentas de [compilação cruzada](#).
 - Configure um espaço de trabalho Linux separado ou use uma ferramenta como o subsistema Windows para Linux (WSL). Você pode usar esse ambiente para executar o Unreal Editor no Linux para criar seu servidor de jogos.


Teste sua integração localmente

Você pode testar sua integração de jogos localmente usando uma GameLift Anywhere frota da Amazon. Essa abordagem é uma prática recomendada para ajudar a isolar problemas diretamente relacionados à integração. Uma Anywhere frota é uma ferramenta útil para executar aplicativos

de teste e cenários de jogos, como iniciar/interromper sessões de jogo e rastrear conexões de jogadores. Você pode criar e testar iterativamente com muito mais rapidez com uma Anywhere frota, que oferece maior visibilidade da atividade de hospedagem.

Consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#) para obter ajuda sobre como usar uma GameLift Anywhere frota da Amazon para testes de integração. O fluxo de trabalho para configurar um ambiente de teste tem a seguinte aparência:

1. Configure um dispositivo local que esteja executando o Linux.
2. Configure uma Anywhere frota. Crie um local personalizado para seu dispositivo local, crie uma Anywhere frota e, em seguida, registre seu dispositivo local como uma computação na frota.
3. Obtenha um token de autenticação para seu servidor de jogos. Seu processo de servidor integrado requer um token para se autenticar com o GameLift serviço da Amazon. Você pode reutilizar o mesmo token para vários processos de servidor em execução simultânea. Essa etapa é necessária somente ao usar uma Anywhere frota para testes de integração.

 Note

Os tokens de autenticação são temporários e devem ser atualizados regularmente. Considere adicionar um script ao pacote de compilação do servidor para solicitar um novo token.

4. Atualize o código do servidor do jogo para Anywhere. Ao ser executado em uma frota Anywhere, o servidor do jogo precisa chamar a ação do SDK do servidor `InitSdk()` ([C++](#)) ([C#](#)) ([Unreal](#)) com os seguintes parâmetros do servidor. Essa etapa é necessária somente ao usar uma Anywhere frota para testes de integração. Depois de adicionar o Amazon GameLift Agent à sua imagem de contêiner, ele manipula esses parâmetros automaticamente.

Como prática recomendada, configure o código do servidor para extrair esses valores das variáveis de ambiente ou dos argumentos do console que você especifica na inicialização.

- `websocketUrl`— Use o valor de `GameLiftServiceSdkEndpoint`, que é retornado da chamada `pararegister-compute`.
- `processId`— Atribua um identificador exclusivo para o processo do servidor.
- `fleetId`— O identificador de frota Anywhere, que é retornado da chamada `paracreate-fleet`.
- `authToken`— Um token de autenticação válido, que é retornado da chamada `paraget-compute-auth-token`.

5. Em sua máquina local, configure o software de criação do servidor de jogos e inicie um processo de servidor.

Se a integração do servidor for bem-sucedida, o processo do servidor chamará a ação do SDK do servidor `InitSDK()` para estabelecer a conexão com o GameLift serviço da Amazon, seguida por uma chamada `ProcessReady()` para notificar o serviço de que está pronto para hospedar uma sessão de jogo.

6. Inicie uma sessão de jogo. Se você integrou seu cliente de jogo para solicitar uma sessão de jogo, poderá usá-lo para solicitar uma nova sessão de jogo. Caso contrário, use o comando AWS CLI. [create-game-session](#) GameLift A Amazon cria um `GameSession` objeto e inicia o processo para iniciar uma nova sessão de jogo.

Se sua integração estiver funcionando, a Amazon GameLift chama um processo de servidor em sua estação de trabalho local para iniciar uma nova sessão de jogo (usando o `onStartGameSession()` callback). Quando uma sessão de jogo está pronta para os jogadores, o processo do servidor é chamado `ActivateGameSession()`. Em resposta, a Amazon GameLift atualiza o `GameSession` status e as informações de conexão para que um cliente do jogo possa se conectar à sessão do jogo e jogar o jogo.

Prepare uma imagem de contêiner com seu software de servidor de jogos

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

O contêiner é o elemento mais básico de uma frota de GameLift contêineres da Amazon. Seu contêiner inclui o servidor do jogo, junto com suas dependências, como SDKs, software, diretórios e arquivos.

Para funcionar em uma frota de contêineres, seu servidor de jogos deve ser executado em Linux e estar integrado ao SDK 5.x do servidor.

Tópicos

- [Configure seu diretório de trabalho](#)
- [Crie sua imagem de contêiner](#)
- [Envie sua imagem de contêiner para o Amazon ECR](#)

Configure seu diretório de trabalho

Seu diretório de trabalho é onde você coloca todos os arquivos necessários para criar sua imagem de contêiner e definir como a Amazon a GameLift executa.

Para configurar seu diretório de trabalho do contêiner

1. Crie o diretório em que você deseja trabalhar com suas imagens de GameLift contêiner da Amazon.

Example

Por exemplo: .

```
[~/]$ mkdir -p work/glc/gamebuild && cd work && find .  
.  
./glc  
./glc/gamebuild
```

2. Clone o [Amazon GameLift Agent](#).

Example

Por exemplo: .

```
[~/work]$ git clone https://github.com/aws/amazon-gamelift-agent.git  
Cloning into 'amazon-gamelift-agent'...
```

3. [Crie o GameLiftAgent usando o Maven](#).

Example

Por exemplo: .

```
[~/work]$ cd amazon-gamelift-agent
```

Example

```
[~/work/amazon-gamelift-agent]$ mvn clean compile assembly:single && \  
mv target ../glc && cd .. && find glc
```

- Adicione um servidor de jogos que tenha sido integrado ao SDK 5.x do servidor, criado e empacotado em um arquivo. .ZIP
- Copie seu .ZIP arquivo para ~/work/glc/gamebuild/ o.

Se você não tem um servidor de jogos SDK 5.x, pode baixar e usar nosso [SimpleServer](#) jogo de amostra para tentar usar uma frota de contêineres.

Example

```
[~/work]$ curl -o glc/gamebuild/SimpleServer.zip \  
'https://ws-assets-prod-iad-r-iad-ed304a55c2ca1aee.s3.us-  
east-1.amazonaws.com/086bb355-4fdc-4e63-8ca7-af7cfc45d4f2/  
AmazonGameLiftSampleServerBinary.zip' &&  
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current  
Dload Upload  Total   Spent    Left  Speed  
100 5140k  100 5140k    0     0  12.3M      0  --:--:--  --:--:--  --:--:-- 12.3M  
glc  
glc/target  
glc/target/GameLiftAgent-1.0.jar  
glc/gamebuild  
glc/gamebuild/SimpleServer.zip
```

Crie sua imagem de contêiner

Seu Dockerfile especifica o ambiente, o software e as instruções para criar seu contêiner.

Para criar seu Dockerfile

- Vá para o glc subdiretório.

Example

```
[~/work]$ cd glc && find  
.  
./target  
./target/GameLiftAgent-1.0.jar  
./gamebuild
```

- Crie e abra um novo Dockerfile.

Example

Por exemplo: .

```
[~/work/glc]$ nano Dockerfile
```

3. Copie de um dos modelos a seguir e cole o conteúdo em seu Dockerfile.

Modelo Dockerfile para seu servidor de jogos

Esse modelo contém as instruções mínimas que um contêiner precisa para ser usado em uma GameLift frota da Amazon. Modifique o conteúdo conforme necessário para seu servidor de jogos.

```
# Base image
# -----
# Add the base image that you want to use over here,
# Make sure to use an image with the same architecture as the
# Instance type you are planning to use on your fleets.
# We require JDK to be installed in the base image, so that
# it can be used to run the &AGS; Agent
FROM public.ecr.aws/amazoncorretto/amazoncorretto:17-amd64
#
# Game build directory
# -----
# Add your game build to gamebuild directory and add the zip file name in the
'GAME_BUILD_ZIP' env variable below.
# The game build provided over here needs to be integrated with gamelift server sdk.
# This template assumes that the game build is in a zip format.
ENV GAME_BUILD_ZIP="<ADD_GAME_BUILD_ZIP_FILE_NAME>" \
#
# Default directory
# -----
# Default directory, the value provided here should be where the game executable
exists.
# Provide this same value as your launch path in RuntimeConfiguration when creating a
fleet.
# Ref: https://docs.aws.amazon.com/gamelift/latest/apireference/
API\_ServerProcess.html
GAME_EXECUTABLE="<ADD NAME OF EXECUTABLE WITHIN THE GAME BUILD>" \
HOME_DIR="/local/game" \
#
# Registered compute in anywhere fleet (not used in container fleets)
```

```
# -----
# Add the name for the registered compute in an anywhere fleet.
# This environment variable is required only for anywhere fleets, but not for
container fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
GAMELIFT_COMPUTE_NAME="<<ADD_COMPUTE_NAME>" \
#
# Default Gamelift Agent jar
# -----
GAMELIFT_AGENT_EXEC="GameLiftAgent-1.0.jar" \
#
# This env variable defines the name of the S3 bucket that stores the GameLift Agent
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
GAMELIFT_AGENT_LOGS_BUCKET_NAME="<<ADD NAME OF GAMELIFT AGENT LOGS S3 BUCKET>" \
#
# -----
# This env variable defines the name of the S3 bucket that stores the game session
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
GAME_SESSION_LOGS_BUCKET_NAME="<<ADD NAME OF GAME SESSION LOGS S3 BUCKET>" \
#
# -----
GAMELIFT_AGENT_LOGS_PATH="/local/game/agentlogs/" \
#
# NOT USED in container fleets - USED in Anywhere fleets
# -----
# Specify the type of compute resource used to host the game servers.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
COMPUTE_TYPE="ANYWHERE" \
#
```

```
# Specify the credential to be used for creating the client.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
CREDENTIAL_PROVIDER="environment-variable"

USER root

# intall dependencies as necessary
RUN yum install -y sudo \
    unzip \
    git \
    shadow-utils \
    iputils \
    tar \
    gcc \
    make \
    openssl-devel \
    zlib-devel \
    vim \
    net-tools \
    nc \
    procps

# Set up the ground for 'gamescale' user
RUN groupadd -r gamescale -g 500 && \
    useradd -u 500 -r -g gamescale -m -s /sbin/nologin -c "Gamescale user" gamescale
&& \
    echo "gamescale ALL=(ALL) NOPASSWD: ALL" | (EDITOR="tee -a" visudo) && \
    mkdir -p $HOME_DIR && \
    mkdir $HOME_DIR/mono && \
    chown -R gamescale:gamescale $HOME_DIR

WORKDIR $HOME_DIR

# extract game build as necessary
COPY ./gamebuild/$GAME_BUILD_ZIP .
RUN unzip ./ $GAME_BUILD_ZIP -d ./

# copy Gamelift Agent jar
COPY ./gameliftAgent/$GAMELIFT_AGENT_EXEC ./
```

```

# Add permissions to game build and gamelift agent jar
RUN chmod +x ./${GAME_EXECUTABLE}
RUN chmod +x ./${GAMELIFT_AGENT_EXEC}

# Check if java is installed on the image, if not then the Agent will not be able
to run
RUN java --version

USER gamescale

ENV PATH="${PATH}:${HOME_DIR}/bin:${JAVA_HOME}"

# Change directory to bin
WORKDIR $HOME_DIR

# check path before starting the container
RUN echo $PATH

# Create logs directory for GameLift Agent & server processes
RUN mkdir logs
RUN mkdir agentlogs

# Start the GameLift Agent
ENTRYPOINT sleep 90 && java -jar ${GAMELIFT_AGENT_EXEC} -ip "192.168.1.1" -gslb
"${GAME_SESSION_LOGS_BUCKET_NAME}" -galb "${GAMELIFT_AGENT_LOGS_BUCKET_NAME}" -galp
"${GAMELIFT_AGENT_LOGS_PATH}" -glc environment-variable

```

Dockerfile para a amostra **SimpleServer**

```

# Base image
# -----
# Add the base image that you want to use over here,
# Make sure to use an image with the same architecture as the
# Instance type you are planning to use on your fleets.
# We require JDK to be installed in the base image, so that
# it can be used to run the &AGS; Agent
FROM public.ecr.aws/amazoncorretto/amazoncorretto:17-amd64
#
# Game build directory
# -----
# Add your game build to gamebuild directory and add the zip file name in the
'GAME_BUILD_ZIP' env variable below.

```

```
# The game build provided over here needs to be integrated with gamelift server sdk.
# This template assumes that the game build is in a zip format.
ENV GAME_BUILD_ZIP="SimpleServer.zip" \
#
# Default directory
# -----
# Default directory, the value provided here should be where the game executable
exists.
# Provide this same value as your launch path in RuntimeConfiguration when creating a
fleet.
# Ref: https://docs.aws.amazon.com/gamelift/latest/apireference/
API\_ServerProcess.html
GAME_EXECUTABLE="GameLiftSampleServer" \
HOME_DIR="/local/game" \
#
# Registered compute in anywhere fleet (not used in container fleets)
# -----
# Add the name for the registered compute in an anywhere fleet.
# This environment variable is required only for anywhere fleets, but not for
container fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
GAMELIFT_COMPUTE_NAME="<ADD_COMPUTE_NAME>" \
#
# Default Gamelift Agent jar
# -----
GAMELIFT_AGENT_EXEC="GameLiftAgent-1.0.jar" \
#
# This env variable defines the name of the S3 bucket that stores the GameLift Agent
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
GAMELIFT_AGENT_LOGS_BUCKET_NAME="<ADD NAME OF GAMELIFT AGENT LOGS S3 BUCKET>" \
#
# -----
# This env variable defines the name of the S3 bucket that stores the game session
logs.
# This S3 bucket should exist in the customer AWS account.
# In order to allow GameLift agent to upload logs to this s3 bucket, customers would
need to
# include s3:PutObject permission in the IAM role provided as instanceRoleArn during
CreateFleet operation.
```

```
# -----
GAME_SESSION_LOGS_BUCKET_NAME="<ADD NAME OF GAME SESSION LOGS S3 BUCKET>" \
#
# -----
GAMELIFT_AGENT_LOGS_PATH="/local/game/agentlogs/" \
#
# NOT USED in container fleets - USED in Anywhere fleets
# -----
# Specify the type of compute resource used to host the game servers.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
COMPUTE_TYPE="ANYWHERE" \
#
# Specify the credential to be used for creating the client.
# This env variable is required only for anywhere fleets, but not for container
fleets.
# If it is set for container fleets, it will be overridden by Gamelift.
#
# -----
CREDENTIAL_PROVIDER="environment-variable"

USER root

# intall dependencies as necessary
RUN yum install -y sudo \
    unzip \
    git \
    shadow-utils \
    iputils \
    tar \
    gcc \
    make \
    openssl-devel \
    zlib-devel \
    vim \
    net-tools \
    nc \
    procps

# Set up the ground for 'gamescale' user
RUN groupadd -r gamescale -g 500 && \
```

```
useradd -u 500 -r -g gamescale -m -s /sbin/nologin -c "Gamescale user" gamescale
&& \
echo "gamescale ALL=(ALL) NOPASSWD: ALL" | (EDITOR="tee -a" visudo) && \
mkdir -p $HOME_DIR && \
mkdir $HOME_DIR/mono && \
chown -R gamescale:gamescale $HOME_DIR

WORKDIR $HOME_DIR

# extract game build as necessary
COPY ./gamebuild/$GAME_BUILD_ZIP .
RUN unzip ./ $GAME_BUILD_ZIP -d ./

# copy Gamelift Agent jar
COPY ./target/$GAMELIFT_AGENT_EXEC ./

# Add permissions to game build and gamelift agent jar
RUN chmod +x ./ $GAME_EXECUTABLE
RUN chmod +x ./ $GAMELIFT_AGENT_EXEC

# Check if java is installed on the image, if not then the Agent will not be able
to run
RUN java --version

USER gamescale

ENV PATH "$PATH:$HOME_DIR/bin:$JAVA_HOME"

# Change directory to bin
WORKDIR $HOME_DIR

# check path before starting the container
RUN echo $PATH

# Create logs directory for GameLift Agent & server processes
RUN mkdir logs
RUN mkdir agentlogs

# Start the GameLift Agent
ENTRYPOINT sleep 90 && java -jar $GAMELIFT_AGENT_EXEC -ip "192.168.1.1" -gslb
"$GAME_SESSION_LOGS_BUCKET_NAME" -galb "$GAMELIFT_AGENT_LOGS_BUCKET_NAME" -galp
"$GAMELIFT_AGENT_LOGS_PATH" -glc environment-variable
```

Note

Nota: Algumas das variáveis de ambiente no Dockerfile podem ser substituídas pelo [ContainerDefinition](#).

Para criar sua imagem de contêiner

1. Crie sua imagem de contêiner.

Se você estiver usando seu próprio servidor SDK 5.x

Você pode especificar o nome do repositório local que desejar.

Example

```
[~/work/glc]$ docker build -t <local repository name>:<optional tag> .
```

Se você estiver usando nossa **SimpleServer** amostra

Example

```
[~/work/glc]$ docker build -t simple-server:version-1 .  
Successfully built 0123456789012  
Successfully tagged simple-server:version-1
```

Note

Nos exemplos a seguir, usamos *simpler-server* como REPOSITORY valor inicial e *version-1* como valor. TAG

2. Veja a lista de imagens e anote IMAGE ID os valores e. REPOSITORY Você precisará deles em um procedimento abaixo.

Example

```
[~/work/glc]$ docker images  
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
```


simple-server version-1 *0123456789012* 14 minutes ago 1.24GB

Envie sua imagem de contêiner para o Amazon ECR

Faça upload da imagem do seu contêiner em um repositório privado no Amazon ECR. Ao criar uma definição de grupo de contêineres, você faz referência a esse local do repositório para que a Amazon GameLift possa tirar um instantâneo da imagem do seu contêiner e usá-la ao implantar uma frota de contêineres.

Note

Se você ainda não tem um repositório privado do Amazon ECR, [crie um](#).

Para obter suas credenciais do Amazon ECR

- Antes de enviar sua imagem de contêiner para o Amazon ECR, você deve adquirir suas AWS credenciais de forma temporária e fornecê-las ao Docker. Obtenha suas credenciais do Amazon ECR para que o Docker possa fazer login.

Example

```
[~/work/glc]$ aws ecr get-login-password --region us-west-2 | docker login --  
username AWS --password-stdin aws_account_id.dkr.ecr.us-west-2.amazonaws.com  
WARNING! Your password will be stored unencrypted in  
/home/user-name/.docker/config.json.  
Configure a credential helper to remove this warning.  
See https://docs.docker.com/engine/reference/commandline/login/#credentials-store  
  
Login Succeeded
```

Para enviar sua imagem de contêiner para o Amazon ECR

1. Copie o URI do [repositório privado do Amazon ECR](#) que você deseja usar.
2. Aplique uma tag Amazon ECR à imagem do seu contêiner.

Example

```
[~/work/glc]$ docker tag <IMAGE ID from above> <Amazon ECR private repository URI>:<optional tag>
```

3. Envie sua imagem de contêiner para o Amazon ECR

Example

```
[~/work/glc]$ docker image push <Amazon ECR private repository URI>
```

Projete uma frota de GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Esses tópicos apresentam as principais decisões que você tomará ao configurar uma frota de GameLift contêineres da Amazon. Suas decisões afetam a forma como você define as configurações para contêineres, grupos de contêineres e frotas.

Tópicos

- [Arquitete sua estrutura de contêineres de frota](#)
- [Defina limites de recursos](#)
- [Designe recipientes essenciais](#)
- [Configurar conexões de rede](#)
- [Configurar verificações de saúde para contêineres](#)
- [Definir dependências de contêineres](#)
- [Configurar uma frota de contêineres](#)

Arquitete sua estrutura de contêineres de frota

Como primeira etapa, identifique o software e os recursos necessários para hospedar seu servidor de jogos, incluindo o seguinte:

- Seu aplicativo de servidor de jogos. O aplicativo deve ser integrado à GameLift funcionalidade de hospedagem da Amazon, incluindo o SDK do servidor versão 5+. Consulte [Integre seu jogo com a Amazon GameLift](#).

- O GameLift agente da Amazon. Esse agente no computador mantém a comunicação com o GameLift serviço da Amazon e gerencia o ciclo de vida de todos os processos do servidor de jogos. Para obter mais detalhes, consulte [Servidores de jogos e o Amazon GameLift Agent](#).
- Software e recursos adicionais conforme necessário. Isso pode incluir o software necessário para executar seus aplicativos de servidor de jogos. O software de suporte comum é usado para registro e monitoramento, segurança, entrega de conteúdo e sincronização de dados.

Em seguida, decida como estruturar seu software e recursos para uma frota de GameLift contêineres da Amazon. A Amazon GameLift usa grupos de contêineres para organizar contêineres. Uma frota sempre tem um grupo de contêineres de réplica e, opcionalmente, pode ter uma frota de contêineres daemon. Para obter mais detalhes, consulte [componentes da frota de contêineres](#).

- Comece projetando seu grupo de contêineres de réplicas. Considere as seguintes diretrizes:
 - Junte seu aplicativo de servidor de jogos e o Amazon GameLift Agent no mesmo contêiner. Faça desse contêiner o único contêiner essencial do grupo de réplicas.
 - Organize todos os outros softwares do seu servidor de jogos em contêineres. Você pode optar por colocar tudo em um único contêiner no grupo de réplicas. Ou você pode optar por criar um ou mais contêineres auxiliares. Alguns motivos para usar sidecars incluem:
 - Para configurar uma sequência de inicialização/desligamento para software individual. Você pode conseguir isso colocando o software em contêineres separados e configurando dependências entre eles.
 - Para definir limites específicos do contêiner para o uso da memória e da CPU.
 - Para especificar diferentes configurações de contêiner para cada contêiner, como comando de inicialização, ponto de entrada, diretório de trabalho, variáveis de ambiente ou verificações de integridade.
- Decida se você precisa de um grupo de contêineres daemon para sua frota. Considere o seguinte:
 - Os contêineres Daemon são normalmente usados para executar processos em segundo plano ou de monitoramento.
 - Os contêineres em um grupo de daemons não são replicados em uma instância de frota. Isso significa que os contêineres em um grupo de daemons não escalam junto com o grupo de contêineres de réplica.
 - Um grupo de daemons pode ter vários contêineres. Você pode designar qualquer contêiner em um grupo de daemons como essencial.

Defina limites de recursos

Para cada grupo de contêineres, determine quanta memória e CPU o grupo precisa para executar seu software. A Amazon GameLift se baseia nessas informações para gerenciar recursos para o grupo de contêineres. Ele também usa essas informações para calcular quantos grupos de contêineres de réplicas uma imagem da frota pode conter. Você também pode definir limites para contêineres individuais.

Defina limites opcionais para contêineres

Definir limites de recursos específicos para contêineres permite que você exerça maior controle sobre como os contêineres individuais podem usar os recursos do grupo. Se você não definir limites específicos para contêineres, todos os contêineres do grupo compartilharão os recursos do grupo. O compartilhamento oferece maior flexibilidade para usar os recursos onde eles são necessários. Também aumenta o potencial de os processos competirem entre si e resultarem em falhas no contêiner.

Defina qualquer uma das `ContainerDefinition` propriedades a seguir para qualquer contêiner.

- `SoftLimit(memória)` — Reserve uma quantidade mínima de memória para uso exclusivo do contêiner. O contêiner sempre tem o valor reservado disponível. Ele pode exceder esse mínimo a qualquer momento, se houver recursos adicionais disponíveis.
- `HardLimit(memória)` — Defina um limite máximo de memória para o contêiner. Se o contêiner exceder esse limite, isso resultará em uma reinicialização.
- `Cpulimite` — Reserve uma quantidade mínima de recursos de CPU para uso exclusivo do contêiner. O contêiner sempre tem o valor reservado disponível. Ele pode exceder esse mínimo a qualquer momento, se houver recursos adicionais disponíveis. (1024 unidades de CPU equivalem a 1 vCPU.)

Definir limites totais de recursos para um grupo de contêineres

Diga à Amazon a GameLift quantidade de recursos de memória e CPU que cada grupo de contêineres precisa. O objetivo é alocar recursos suficientes para otimizar o desempenho do servidor de jogos. A Amazon GameLift usa esses limites para calcular como empacotar grupos de contêineres de réplicas em uma instância de frota. Você também os usará ao escolher um tipo de instância para uma frota de contêineres.

Calcule a memória total e a CPU necessárias para todos os processos em cada contêiner em um grupo. Considere o seguinte:

- Quais processos são executados em todos os contêineres do grupo de contêineres? Some os recursos necessários para esses processos.
- Quantos processos simultâneos de servidor de jogos você planeja executar em cada grupo de contêineres? Você define esse valor como parte da configuração de tempo de execução de uma frota, mas precisa planejar memória suficiente para eles aqui (consulte [Otimizar sua configuração de tempo de execução](#)).

Com base em sua estimativa dos requisitos do grupo de contêineres, defina as seguintes `ContainerGroupDefinition` propriedades:

- `TotalMemoryLimit`— Defina um limite máximo de memória para o grupo de contêineres. Todos os contêineres do grupo compartilham a memória alocada. Se você definir limites de contêineres individuais, o limite total de memória deverá ser:
 - igual ou maior que a soma de todos os limites de memória flexível do contêiner
 - igual ou maior que o limite máximo de memória rígida para um contêiner no grupo
- `TotalCpuLimit` — Defina um limite máximo de CPU para o grupo de contêineres. Todos os contêineres do grupo compartilham os recursos de CPU alocados. Se você definir limites de contêineres individuais, o limite total de CPU deverá ser:
 - igual ou maior que a soma de todos os limites de CPU do contêiner. Como prática recomendada, considere definir esse valor para dobrar a soma dos limites de CPU do contêiner.

Exemplo de cenário

Digamos que estamos definindo um grupo de contêineres de réplica com os três contêineres a seguir:

- O contêiner A é nosso contêiner de réplica essencial. Ele executa processos do servidor de jogos e o Amazon GameLift Agent. Estimamos os requisitos de recursos para um servidor de jogos em 512 MiB e 1024 de CPU. Planejamos que o contêiner execute 10 processos do servidor. Como esse contêiner executa nosso software mais importante, definimos uma reserva de memória temporária de 6144 MiB e nenhum limite de memória rígida ou limite de reserva de CPU.
- O contêiner B executa software de suporte com requisitos de recursos estimados em 1024 MiB e 1536 de CPU. Definimos um limite de reserva de memória flexível de 1024 MiB, um limite de memória rígida de 2048 MiB e um limite de reserva de CPU de 1024 CPUs.
- O contêiner C executa registros não críticos e outros utilitários de monitoramento. Definimos um limite de memória rígida de 512 MiB e um limite de reserva de CPU de 512 CPUs.

Usando essas informações, definimos os seguintes limites totais para o grupo de contêineres:

- Limite total de memória: 7680 MiB. Esse valor excede (1) a soma dos limites de memória flexível (6144+1024 MiB) e (2) o limite máximo de memória rígida (1024 MiB).
- Limite total de CPU: 13312 CPU. Esse valor excede a soma do limite de CPU (1024+512 CPU).

Designe recipientes essenciais

Para cada recipiente, designe o recipiente como essencial ou não essencial. Todos os grupos de contêineres devem ter pelo menos um contêiner essencial. O contêiner essencial faz o trabalho crítico do grupo de contêineres, como hospedar seus servidores de jogos. Espera-se que o contêiner essencial esteja sempre funcionando. Se falhar, todo o grupo de contêineres será reiniciado.

- O grupo de réplicas de contêineres da sua frota pode ter exatamente um contêiner essencial. Esse contêiner executa o Amazon GameLift Agent e os processos do servidor de jogos que ele gerencia.
- Se sua frota tiver um grupo de contêineres daemon, você poderá designar vários contêineres essenciais. Torne um contêiner daemon essencial se você quiser que uma falha no contêiner solicite a reinicialização do grupo de contêineres.

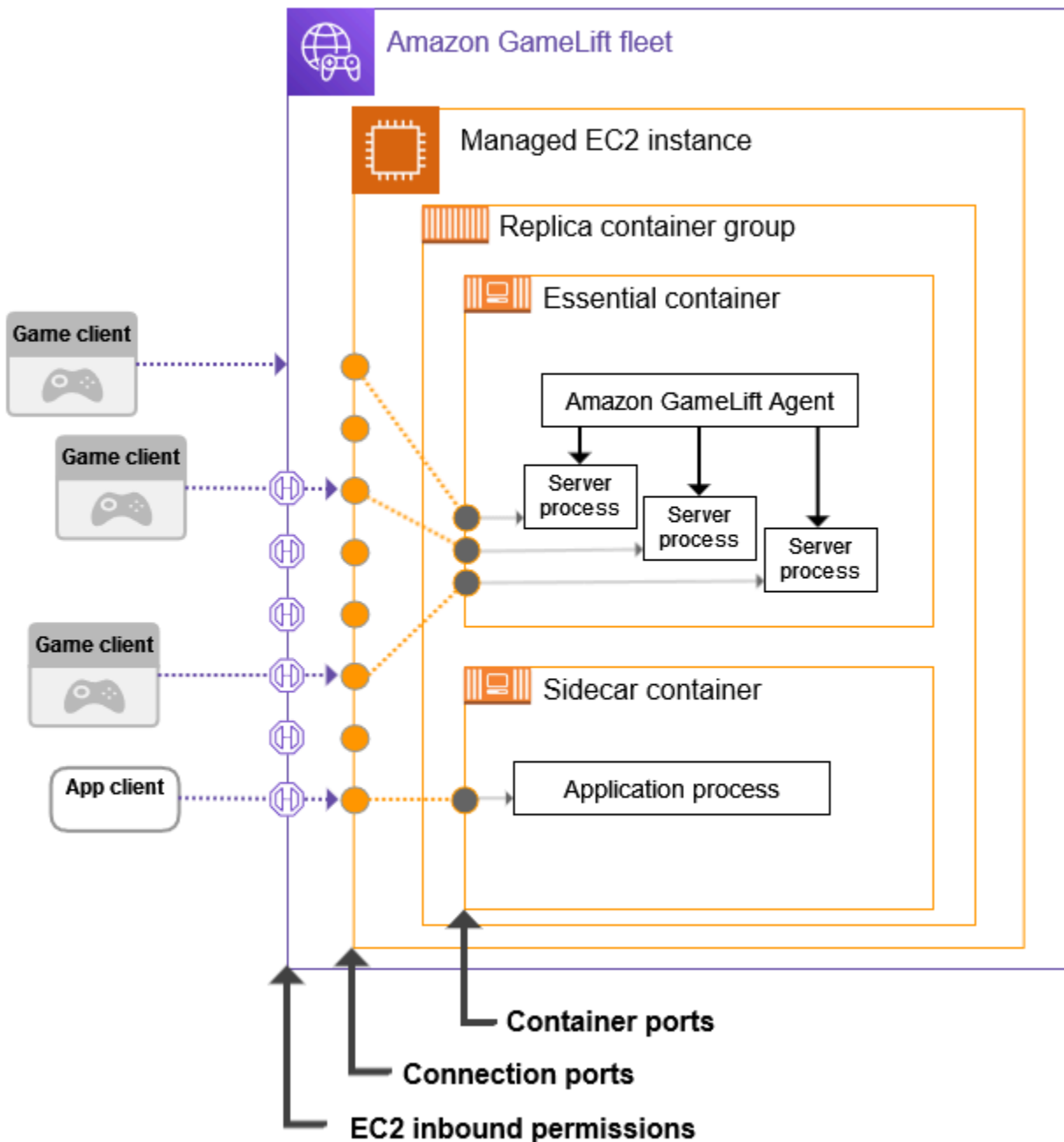
Defina `Essential` a `ContainerDefinition` propriedade como verdadeira ou falsa para cada contêiner.

Configurar conexões de rede

Você pode estabelecer acesso à rede para permitir que o tráfego externo se conecte a qualquer contêiner em uma frota de contêineres. Por exemplo, você deve estabelecer conexões de rede com o contêiner que executa os processos do servidor de jogos, para que os clientes do jogo possam entrar e jogar seu jogo. Clientes de jogos se conectam aos servidores de jogos usando portas e endereços IP.

Em uma frota de contêineres, a conexão entre cliente e servidor não é direta. Internamente, um processo em um contêiner escuta em uma porta de contêiner. Externamente, o tráfego de entrada se conecta a uma instância de frota usando uma porta de conexão. A Amazon GameLift mantém os mapeamentos entre as portas internas do contêiner e as portas de conexão externas, para que o tráfego de entrada seja roteado para o processo correto na instância.

GameLift A Amazon fornece uma camada extra de controle para suas conexões de rede. Cada frota de contêineres tem uma configuração de permissões de entrada, que permite controlar o acesso a cada porta de conexão externa. Você não pode alterar as configurações de porta de uma frota existente, mas pode permitir ou restringir o acesso conforme necessário ajustando as permissões de entrada. Por exemplo, você pode remover as permissões de todas as portas de conexão para desligar todo o acesso aos contêineres da frota.



Definir intervalos de portas de contêineres

Configure uma definição de contêiner com portas de contêiner suficientes para qualquer processo que precise de acesso externo. Alguns contêineres não precisarão de portas. Outros devem ter portas suficientes para atribuir uma a cada processo que precise de uma.

Seu grupo essencial de contêineres de réplicas, que executa seus servidores de jogos, precisa de uma porta para cada processo de servidor de jogos em execução simultânea (conforme configurado na frota). `RuntimeConfiguration` O processo do servidor do jogo escuta a porta atribuída e a reporta à Amazon GameLift.

Ao criar uma definição de grupo de contêineres, defina um intervalo de portas de contêiner para cada contêiner que precisa de acesso à rede (consulte [ContainerDefinitionInput: PortConfiguration](#)). Certifique-se de que o intervalo seja grande o suficiente para atribuir uma porta a cada processo que precise de uma. Os processos devem ter números de porta atribuídos na configuração da porta do contêiner.

Definir intervalos de portas de conexão

Configure sua frota de contêineres com um conjunto de portas de conexão. As portas de conexão fornecem acesso externo às instâncias da frota que estão executando seus contêineres. A Amazon GameLift atribui portas de conexão e as mapeia para portas de contêineres conforme necessário.

Ao criar uma frota de contêineres, defina um intervalo de portas de conexão (consulte [ContainerGroupsConfiguration: ConnectionPortRange](#)). Certifique-se de que o intervalo tenha portas suficientes para mapear para cada porta de contêiner em uma instância de frota. Para calcular as portas de conexão mínimas necessárias, use a seguinte fórmula:

```
[Total number of container ports defined for containers in the replica container group] * [Number of replica container groups per instance] + [Total number of container ports defined for containers in the daemon container group]
```

Como prática recomendada, dobre o número mínimo de portas de conexão.

Note

O número de portas de conexão pode potencialmente limitar o número de grupos de contêineres de réplicas por instância. Se uma frota tiver apenas portas de conexão

suficientes para um grupo de contêineres de réplica por instância, a Amazon GameLift implantará somente um grupo de contêineres de réplica, mesmo que as instâncias tenham capacidade computacional suficiente para vários grupos de contêineres de réplica.

Defina permissões de entrada

As permissões de entrada controlam o acesso externo a uma frota de contêineres especificando quais portas de conexão devem ser abertas para o tráfego de entrada. Você pode usar essa configuração para ativar e desativar o acesso à rede de uma frota conforme necessário.

[Ao criar uma frota de contêineres, defina um conjunto de permissões de entrada \(CreateFleetconsulte:EC2\). InboundPermissions](#) Defina as propriedades da porta de permissão de entrada para incluir alguns ou todos os valores nas configurações da porta de conexão da frota. Para alterar as permissões de entrada em uma frota de contêineres existente, ligue [UpdateFleetPortSettings](#).

Exemplo de cenário

Este exemplo ilustra como definir todas as três propriedades de conexão de rede.

- O grupo de contêineres de réplicas da nossa frota tem 1 contêiner, que executa os processos do servidor do jogo. A confirmação do tempo de execução diz ao contêiner que execute 10 processos simultâneos do servidor de jogos.

Na definição do grupo de contêineres de réplica, definimos o `PortConfiguration` parâmetro para esse contêiner da seguinte forma:

```
"PortConfiguration": {
  "ContainerPortRanges": [ { "FromPort": 10, "ToPort": 20, "Protocol": "TCP" } ]
}
```

- Nossa frota também tem um grupo de contêineres daemon com 1 contêiner. Ele tem 1 processo que precisa de acesso à rede. Na definição do grupo de contêineres daemon, definimos o `PortConfiguration` parâmetro para esse contêiner da seguinte forma:

```
"PortConfiguration": {
  "ContainerPortRanges": [ { "FromPort": 25, "ToPort": 25, "Protocol": "TCP" } ] }
```

- Nossa frota é configurada com 3 grupos de contêineres de réplicas por instância da frota. Com essas informações, podemos usar a fórmula para calcular o número de portas de conexão necessárias:
 - Mínimo: 31 portas [10 portas de contêiner de réplica * 3 grupos de contêineres de réplica por instância + 1 porta de contêiner daemon]
 - Melhor prática: 62 portas [portas mínimas* 2]

Ao criar a frota de contêineres, definimos o `ConnectionPortRange` parâmetro da `ContainerGroupsConfiguration` seguinte forma:

```
"ConnectionPortRange": { "FromPort": 1010, "ToPort": 1071 }
```

- Queremos permitir o acesso a todas as portas de conexão disponíveis. Ao criar a frota de contêineres, definimos o `EC2InboundPermissions` parâmetro da seguinte forma:

```
"EC2InboundPermissions": [  
  {"FromPort": 1010, "ToPort": 1071, "IpRange": "10.24.34.0/23", "Protocol":  
  "TCP"} ]
```

Configurar verificações de saúde para contêineres

Um contêiner é reiniciado automaticamente se ocorrer uma falha no terminal e parar de funcionar. Se o contêiner for essencial, todo o grupo de contêineres será reiniciado.

Você pode definir critérios personalizados adicionais para medir a integridade do contêiner e usar uma verificação de saúde para testar esses critérios. Para configurar uma verificação de integridade do contêiner, você pode defini-la em uma imagem de contêiner docker ou em sua definição de contêiner. Se você definir uma verificação de saúde na definição do contêiner, ela substituirá todas as configurações na imagem do contêiner.

Defina verificações de saúde opcionais com base no tipo de contêiner da seguinte forma:

- Para um contêiner de réplica essencial, não configure verificações de saúde. O Amazon GameLift Agent gerencia automaticamente os relatórios de saúde desse contêiner.
- Para contêineres de réplica não essenciais e qualquer contêiner daemon, você pode, opcionalmente, definir parâmetros de verificação de integridade.

Defina as seguintes `ContainerDefinition` propriedades para uma verificação de integridade do contêiner:

- `Command`— Forneça um comando que verifique alguns aspectos da integridade do contêiner. Você decide quais critérios usar para medir a saúde. O comando deve resultar em um valor de saída de 1 (não íntegro) ou 0 (íntegro).
- `StartPeriod`— Especifique um atraso inicial antes que as falhas na verificação de integridade comecem a ser contadas. Esse atraso dá ao contêiner tempo para inicializar seus processos.
- `Interval`— Decida com que frequência executar o comando de verificação de integridade. Com que rapidez você deseja detectar e resolver uma falha no contêiner?
- `Timeout`— Decida quanto tempo esperar pelo sucesso ou pela falha antes de tentar novamente o comando de verificação de integridade. Quanto tempo o comando de verificação de integridade deve levar para ser concluído?
- `Retries`— Quantas vezes o comando de verificação de integridade deve ser repetido antes de registrar uma falha?

Definir dependências de contêineres

Em cada grupo de contêineres, você pode definir dependências entre contêineres com base no status do contêiner. Uma dependência afeta quando o contêiner dependente pode ser iniciado ou encerrado com base no status de outro contêiner.

Um caso de uso importante para dependências é criar sequências de inicialização e desligamento para o grupo de contêineres.

Por exemplo, você pode querer que o contêiner A comece primeiro e seja concluído com êxito antes do início dos contêineres B e C. Para conseguir isso, primeiro crie uma dependência para o contêiner B no contêiner A, com a condição de que o contêiner A seja concluído com êxito. Em seguida, crie uma dependência para o contêiner C no contêiner A com a mesma condição. As sequências de inicialização ocorrem na ordem inversa para o desligamento.

Configurar uma frota de contêineres

Ao criar uma frota de contêineres, considere os seguintes pontos de decisão. A maioria desses pontos depende da arquitetura e da configuração do contêiner.

Decida onde você deseja implantar sua frota

Em geral, você deseja implantar suas frotas geograficamente perto de seus jogadores para minimizar a latência. Você pode implantar sua frota de contêineres em qualquer plataforma Região da AWS que a Amazon GameLift ofereça suporte. Se você quiser implantar o mesmo servidor de jogo em outras localizações geográficas, você pode adicionar locais remotos à frota, incluindo Zonas Regiões da AWS Locais. Para uma frota com vários locais, você pode ajustar a capacidade de forma independente em cada localização da frota. Para obter mais informações sobre as localizações de frotas suportadas, consulte [Locais GameLift de hospedagem da Amazon](#).

Escolha um tipo e tamanho de instância para sua frota

A Amazon GameLift oferece suporte a uma ampla variedade de tipos de instâncias do Amazon EC2, todas disponíveis para uso com uma frota de contêineres. A disponibilidade e o preço do tipo de instância variam de acordo com o local. Você pode ver uma lista dos tipos de instância compatíveis, filtrados por localização, no GameLift console da Amazon (em Recursos, Instância e cotas de serviços).

Ao escolher um tipo de instância, primeiro considere a família de instâncias. As famílias de instâncias oferecem várias combinações de recursos de CPU, memória, armazenamento e rede. Obtenha mais informações sobre [famílias de instâncias do EC2](#). Em cada família, você tem uma variedade de tamanhos de instância para escolher. Considere os seguintes problemas ao selecionar um tamanho de instância:

- Qual é o tamanho mínimo de instância que pode suportar sua carga de trabalho? Use essas informações para eliminar qualquer tipo de instância que seja muito pequeno.
- Quais tamanhos de tipo de instância são adequados para sua arquitetura de contêiner? O ideal é escolher um tamanho que possa acomodar várias cópias do seu grupo de contêineres de réplica com o mínimo de espaço desperdiçado.
- Qual granularidade de escalabilidade faz sentido para seu jogo? A capacidade da frota de escala envolve adicionar ou remover instâncias, e cada instância representa a capacidade de hospedar um número específico de sessões de jogo. Considere a capacidade que você deseja adicionar ou remover com cada instância. Se a demanda dos jogadores variar milhares de minuto a minuto, talvez faça sentido usar instâncias muito grandes que possam hospedar centenas ou milhares de sessões de jogo. Por outro lado, você pode preferir um controle de escalabilidade mais refinado com tipos de instância menores.
- Há economia de custos disponível com base no tamanho? Você pode descobrir que o custo de certos tipos de instância varia de acordo com o local devido à disponibilidade.

Otimize sua configuração de tempo de execução

A configuração de tempo de execução de uma frota é um conjunto de instruções sobre como executar processos de servidor para hospedagem de sessões de jogos. Essas instruções são implementadas pelo GameLift agente da Amazon em cada grupo de contêineres de réplica na frota.

A configuração de tempo de execução de uma frota determina quantos processos de servidor são executados simultaneamente em cada grupo de contêineres de réplica. Essa configuração afeta a forma como você calcula os limites de recursos do seu grupo de contêineres e como escolhe um tipo de instância para sua frota. Você precisa equilibrar esses três elementos ao projetar sua frota.

Para obter mais informações sobre como usar as configurações de tempo de execução, consulte [Gerencie como o Amazon GameLift lança servidores de jogos](#).

Defina outras configurações opcionais da frota

Você pode usar os seguintes recursos opcionais ao configurar uma frota de contêineres:

- Configure seus servidores de jogos para acessar outros AWS recursos. Consulte [Comunique-se com outros recursos AWS de suas frotas](#).
- Proteja as sessões de jogo com jogadores ativos de serem encerradas prematuramente durante um evento de redução de escala.
- Limite o número de sessões de jogo que um indivíduo pode criar na frota em um período limitado de tempo.

Crie definições de grupos de contêineres para uma frota de GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Uma definição de grupo de contêineres descreve como implantar seus aplicativos de servidor de jogos em contêineres em uma frota de contêineres. É um plano que identifica o conjunto de contêineres a serem executados na frota e como operá-los. Ao criar uma frota de contêineres, você especifica as definições do grupo de contêineres a serem implantadas na frota. Para obter mais informações sobre grupos de contêineres, consulte [componentes da frota de contêineres](#).

Antes de começar

Complete as seguintes tarefas:

- Crie uma arquitetura de contêiner para hospedar seus servidores de jogos. Consulte [Projete uma frota de GameLift contêineres da Amazon](#).
- Planeje as definições de contêineres a serem incluídas no grupo de contêineres. Se você estiver usando a AWS CLI, crie sua definição de contêiner em um arquivo JSON.
- Envie as imagens finais do contêiner para um registro do Amazon Elastic Container Registry (Amazon ECR) no Região da AWS mesmo local em que você planeja criar o grupo de contêineres. A Amazon GameLift armazena um instantâneo de cada imagem no momento em que você cria a definição do grupo de contêineres e usa a cópia ao implantar em uma frota de contêineres. Consulte [Prepare uma imagem de contêiner com seu software de servidor de jogos](#).
- Verifique se seu AWS usuário tem permissões do IAM para acessar o repositório Amazon ECR. Consulte [Gerenciar permissões de usuário para a Amazon GameLift](#). No mínimo, você precisa de permissões para as seguintes ações:
 - `ecr:DescribeImages`
 - `ecr:BatchGetImage`
 - `ecr:GetDownloadUrlForLayer`

Clonar uma definição de grupo de contêineres

Você pode usar o GameLift console da Amazon para clonar uma definição de grupo de contêineres existente.

Para clonar um grupo de contêineres

1. No [GameLift console da Amazon](#), vá para o painel de navegação esquerdo e escolha Grupos de contêineres.
2. Na página de listagem de grupos de contêineres, selecione o grupo de contêineres existente que você deseja clonar. Depois de selecionar um grupo de contêineres, o botão Clonar fica ativo.
3. Escolha Clonar. Essa ação abre o assistente de criação de grupos de contêineres com configurações pré-preenchidas.
4. Insira um novo nome para o grupo de contêineres clonados. O grupo de contêineres na mesma região deve ter nomes exclusivos.

5. Percorra o grupo de contêineres e as páginas de definição de contêineres, revise e crie o novo grupo de contêineres.

Criar uma definição de grupo de contêineres de réplica

Um grupo de contêineres de réplicas gerencia seu software de servidor de jogos. Um grupo de contêineres de réplica tem pelo menos um contêiner que executa o Amazon GameLift Agent e os processos do seu servidor de jogos. O grupo pode ter contêineres “auxiliares” adicionais para executar o software de suporte.

Este tópico descreve como criar uma definição de grupo de contêineres usando o GameLift console da Amazon ou as ferramentas de AWS CLI. Para obter informações mais detalhadas sobre como definir configurações de grupos de contêineres, consulte [Projete uma frota de GameLift contêineres da Amazon](#)

Console

No [GameLift console da Amazon](#), selecione Região da AWS onde você deseja criar o grupo de contêineres.

Abra a barra de navegação esquerda do console e escolha Grupos de contêineres. Na página Grupos de contêineres, escolha Criar grupo de contêineres.

Etapa 1: Defina os detalhes do grupo.

1. Insira um nome de definição de grupo de contêineres. Esse nome deve ser exclusivo para a região Conta da AWS e. No console, as definições de grupo são listadas por nome, portanto, pode ser útil atribuir rótulos significativos.
2. Selecione a estratégia de agendamento da réplica.
3. Em Limite total de memória, insira a memória máxima disponível para o grupo de contêineres. Para obter ajuda no cálculo desse valor, consulte [Defina limites de recursos](#).
4. Em Limite total de CPU, insira a potência máxima de computação disponível para o grupo de contêineres. Para obter ajuda no cálculo desse valor, consulte [Defina limites de recursos](#).

Etapa 2: adicionar definições de contêiner.

Defina o contêiner com seu aplicativo de servidor de jogos e o Amazon GameLift Agent. Este é o seu recipiente de réplica essencial.

1. Forneça um nome de definição de contêiner. Cada contêiner definido para o grupo deve ter um valor de nome exclusivo.
2. Identifique o URI da imagem do Amazon ECR da imagem do contêiner. Insira qualquer um dos seguintes formatos:
 - Somente URI da imagem: [Conta da AWS].dkr.ecr.[Região da AWS].amazonaws.com/[repository ID]
 - URI da imagem + resumo: [Conta da AWS].dkr.ecr.[Região da AWS].amazonaws.com/[repository ID]@[digest]
 - URI da imagem + tag: [Conta da AWS].dkr.ecr.[Região da AWS].amazonaws.com/[repository ID]:[tag]
3. Para o contêiner Essential, Sim é selecionado automaticamente para a primeira definição do contêiner. Se você adicionar outra definição de contêiner, poderá ativar ou desativar essa configuração para cada definição. Para obter mais detalhes, consulte [Designe recipientes essenciais](#).
4. Defina um ou mais intervalos de portas internas do contêiner. Esse contêiner hospeda seus servidores de jogos, então defina um intervalo com portas suficientes para que cada processo do servidor seja executado no grupo de contêineres. Para obter mais detalhes, consulte [Configurar conexões de rede](#).
5. As configurações opcionais Overrides e as variáveis Environment permitem que você especifique valores a serem passados para o contêiner na inicialização. Os valores definidos aqui substituem todas as configurações que já estão na imagem do contêiner.
6. Defina limites de contêiner opcionais para gerenciar a alocação de recursos para esse contêiner. Para obter mais detalhes, consulte [Defina limites de recursos](#).
7. Defina recipientes adicionais não essenciais conforme necessário:
 - Forneça um nome de definição de contêiner e um URI de imagem ECR. Os contêineres não essenciais não devem executar o Amazon GameLift Agent.
 - Defina um intervalo de portas de contêiner interno somente se os contêineres tiverem processos que precisem de acesso à rede.
 - Opcionalmente, configure uma verificação de saúde para o contêiner. Quando um contêiner não essencial falha em uma verificação de integridade, ele solicita a reinicialização somente do contêiner com falha.
 - Opcionalmente, defina substituições, variáveis de ambiente e limites de alocação de recursos conforme necessário.

Etapa 3: configurar dependências.

Se você tiver mais de um contêiner na definição do grupo de contêineres, poderá definir dependências entre eles. Use dependências para configurar sequências de inicialização e desligamento com base na condição do contêiner. Para obter mais detalhes, consulte [Definir dependências de contêineres](#).

1. Identifique o nome do contêiner ao qual você deseja adicionar uma dependência. Esse contêiner não inicia até que a condição de dependência seja satisfeita.
2. Identifique o nome e a condição do contêiner de dependência. Esse contêiner deve atender às condições antes que o contêiner dependente possa ser iniciado.
3. Defina dependências adicionais conforme necessário. Você pode criar várias dependências para qualquer contêiner. Evite criar dependências circulares.

Etapa 4: revise e crie.

1. Revise todas as configurações de definição do grupo de contêineres. Você não pode alterar a configuração de uma definição de grupo de contêineres após sua criação. Use Editar para fazer alterações em qualquer seção, incluindo cada uma das definições de contêiner para o grupo.
2. Quando terminar de revisar, escolha Criar.

Se sua solicitação for bem-sucedida, o console exibirá a página de detalhes do novo recurso de definição de grupo de contêineres. Inicialmente `COPYING`, o status é: a Amazon GameLift começa a tirar fotos de todas as imagens de contêineres do grupo. Quando essa fase for concluída, o status da definição do grupo de contêineres será alterado para `READY`. Uma definição de grupo de contêineres deve estar em `READY` status antes que você possa criar uma frota de contêineres com ela.

AWS CLI

Ao usar a AWS CLI para criar uma definição de grupo de contêineres, mantenha suas configurações de definição de contêiner em um arquivo separado. JSON Você pode referenciar o arquivo no comando da CLI. Consulte [Criar um JSON arquivo de definição de contêiner](#) para obter exemplos de esquemas.

Criar uma definição de grupo de contêineres

Para criar uma nova definição de grupo de contêineres, use o `create-container-group-definition` comando CLI. Para obter mais informações sobre esse comando, consulte [create-container-group-definition](#) na Referência de Comandos da AWS CLI.

Example : Grupo de contêineres de réplica

Este exemplo ilustra uma solicitação para a definição de um grupo de contêineres de réplica. A estrutura de comando para criar definições de réplicas e grupos de daemons é essencialmente idêntica. Detalhes específicos de cada tipo de grupo são descritos nas definições de contêineres individuais.

Este exemplo pressupõe que você tenha criado um arquivo JSON com as definições de contêiner desse grupo.

```
aws gamelift create-container-group-definition \  
  --name MyAdventureGameContainerGroup \  
  --operating-system AMAZON_LINUX_2023 \  
  --scheduling-strategy REPLICA \  
  --total-memory-limit 4096 \  
  --total-cpu-limit 1024 \  
  --container-definitions file://SimpleServer.json
```

Criar um **JSON** arquivo de definição de contêiner

Ao criar uma definição de grupo de contêineres, você também define os contêineres para o grupo. Uma definição de contêiner especifica o repositório Amazon ECR onde a imagem do contêiner é armazenada e configurações opcionais para portas de rede, limites de uso de CPU e memória e outras configurações. Recomendamos criar um único JSON arquivo com as configurações de todos os contêineres em um grupo de contêineres. A manutenção de um arquivo é útil para armazenar, compartilhar e rastrear versões dessas configurações críticas. Se você usar a AWS CLI para criar suas definições de grupo de contêineres, poderá referenciar o arquivo no comando.

Para criar uma definição de contêiner

1. Crie e abra um novo `.JSON` arquivo. Por exemplo: .

```
[~/work/glc]$ vim SimpleServer.json
```

2. Crie uma definição de contêiner separada para cada um dos contêineres do grupo. Copie o conteúdo de exemplo a seguir e modifique-o conforme necessário para seus contêineres. Para obter detalhes sobre a sintaxe de uma definição de contêiner, consulte [ContainerDefinitionInput](#) Amazon GameLift API Reference.
3. Salve o arquivo localmente para que você possa consultá-lo em um comando da AWS CLI.

Exemplo: definição essencial de contêiner de réplica

Example

Este exemplo descreve o contêiner essencial para seu grupo de contêineres de réplica. O contêiner de réplica essencial inclui seu aplicativo de servidor de jogos, o Amazon GameLift Agent, e pode incluir outro software de suporte para sua hospedagem de jogos. A definição deve incluir um nome, URI de imagem e uma configuração de porta. Este exemplo também define alguns limites de recursos específicos do contêiner.

```
[
  {
    "ContainerName": "SimpleServer",
    "ImageUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/gl-containers:complex-server",
    "Essential": true,
    "Cpu": 256,
    "MemoryLimits": {
      "HardLimit": 128
    },
    "PortConfiguration": {
      "ContainerPortRanges": [
        {
          "FromPort": 2000,
          "Protocol": "TCP",
          "ToPort": 2100
        }
      ]
    }
  }
]
```

Crie uma frota de GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Depois de criar suas definições de grupo de contêineres, use o [GameLift console da Amazon](#) ou o AWS Command Line Interface (AWS CLI) para criar uma frota de contêineres.

Depois de criar uma nova frota, o status da frota passa por vários estágios à medida que a Amazon GameLift implanta seus grupos de contêineres em cada instância da frota e inicia os servidores do jogo. Quando a frota atingir o status ACTIVE, ela estará pronta para sediar sessões de jogo. Para obter ajuda com problemas de criação de frotas, consulte [Depurar problemas com a frota do Amazon GameLift](#).

Console

No [GameLift console da Amazon](#), selecione Região da AWS onde você deseja criar a frota. As definições do grupo de contêineres devem estar na mesma região em que você deseja criar a frota.

Abra a barra de navegação esquerda do console e escolha Fleets. Na página Frotas, escolha Criar frota.

Etapa 1: escolha o tipo de computação

- Escolha o tipo de computação de contêineres.

Etapa 2: Definir detalhes da frota

1. Na seção Detalhes da frota, insira o nome e a descrição da frota.
2. Na seção Detalhes do grupo de contêineres, identifique os grupos de contêineres a serem implantados na frota. Você deve adicionar um grupo de contêineres de réplica. Opcionalmente, você pode adicionar um grupo de contêineres daemon. Cada grupo deve estar em status READY.
3. Defina o intervalo de portas de conexão para a frota. Para obter mais detalhes, consulte [Configurar conexões de rede](#).
4. Opcionalmente, especifique as réplicas desejadas por instância a serem implantadas. Você pode especificar um número desejado ou deixar a Amazon GameLift calcular o número máximo possível. Se você especificar um número desejado maior do que o máximo calculado, a criação da frota falhará. Você não pode alterar essa configuração após a criação

da frota. Para obter mais detalhes sobre a embalagem de grupos de contêineres de réplicas, consulte [Conceitos principais](#).

5. (Opcional) Em Detalhes adicionais:
 - a. Para a função Instance, especifique uma função do IAM que autorize os aplicativos em sua compilação de jogo a acessar outros AWS recursos em sua conta. Para ter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#). Para criar uma frota com uma função de instância, sua conta precisa ter a permissão de PassRole do IAM. Para ter mais informações, consulte [Exemplos de permissões do IAM do Amazon GameLift](#).
 - b. Em Grupo de métricas, insira o nome de um grupo de métricas de frota novo ou existente. É possível agregar as métricas de várias frotas, adicionando-as ao mesmo grupo de métricas.

Etapa 3: definir detalhes da instância

1. Em Implantação de instâncias, selecione um ou mais locais remotos para implantar instâncias. A região de origem é selecionada automaticamente (essa é a região na qual você está criando a frota). Caso selecione locais adicionais, as instâncias da frota também serão implantadas nesses locais.

Important

Para usar regiões que não estão habilitadas por padrão, ative-as em seu Conta da AWS.

- As frotas com regiões não habilitadas que você criou antes de 28 de fevereiro de 2022 não são afetadas.
- Para criar novas frotas com vários locais ou atualizar frotas com vários locais existentes, primeiro ative todas as regiões que você escolher usar.

Para obter mais informações sobre regiões que não estão habilitadas por padrão e como habilitá-las, consulte [Gerenciando Regiões da AWS](#) no Referência geral da AWS.

2. Selecione uma configuração de instância para a frota. O console calcula automaticamente a vCPU e a memória mínimas necessárias (com base nos limites totais definidos para cada

grupo de contêineres). Ele filtra a lista completa dos tipos de instância disponíveis com base nos requisitos de recursos e nos locais que você inseriu. Você pode adicionar filtros adicionais conforme necessário.

Para obter mais informações sobre a escolha de um tipo de instância, consulte [Configurar uma frota de contêineres](#). O tamanho do tipo de instância que você escolher afetará a forma como os grupos de contêineres de réplicas são agrupados em cada instância da frota. Dependendo da sua escolha, considere revisar sua configuração para ver as réplicas desejadas por instância.

Etapa 4: configurar o tempo de execução

A configuração do tempo de execução determina como os processos do servidor de jogos são iniciados e executados. Essas instruções são passadas para o Amazon GameLift Agent, que as implementa da mesma forma em cada grupo de contêineres de réplica. Você pode atualizar a configuração de tempo de execução de uma frota ligando para [UpdateRuntimeConfiguration](#).

1. Em Launch path, insira o caminho para o executável de um jogo.
2. (Opcional) Em Parâmetros de inicialização, insira as informações a serem passadas para o executável do jogo como um conjunto de parâmetros da linha de comando.
3. Especifique o número de processos simultâneos a serem mantidos em execução em cada grupo de contêineres de réplica. Analise as GameLift [cotas](#) da Amazon sobre o número de processos do servidor por instância. Limites de processos de servidor simultâneos por instância se aplicam ao total de processos simultâneos para todas as configurações. Se você configurar a frota para exceder o limite, ela não poderá ser ativada.
4. Defina limites opcionais para ativações simultâneas de sessões de jogo. Essas configurações permitem limitar a quantidade de recursos consumidos ao iniciar uma nova sessão de jogo. As ativações das sessões de jogo podem ter um impacto no desempenho das sessões de jogo existentes.
5. Defina as configurações de porta do EC2 para permitir que o tráfego externo tenha acesso aos processos que estão sendo executados na frota. Especifique alguns ou todos os números de portas de conexão definidos para a frota. Você não precisa definir essas portas ao criar a frota, mas sem elas, nenhum tráfego pode se conectar aos seus servidores de jogos. Para atualizar as configurações de porta de uma frota posteriormente, ligue [UpdateFleetPortSettings](#)

6. Em Configurações de recursos da sessão de jogo, configure os seguintes recursos opcionais:
 - a. Ative ou desative a política de proteção de escalonamento de jogos. Com a proteção ativada, a Amazon GameLift não encerrará instâncias durante um evento de redução se estiver hospedando uma sessão de jogo ativa.
 - b. Defina um limite máximo de criação de recursos para limitar o número de sessões de jogo que um jogador pode criar durante um período de tempo especificado.

Etapa 5: configurar tags

- (Opcional) Adicione tags à compilação inserindo pares de Chave e Valor. Selecione a opção Avançar para continuar com a revisão da criação da frota.

Etapa 6: Revise e crie.

- Revise as configurações da sua frota.

Atualize os metadados da frota e a configuração a qualquer momento, independentemente do status da frota. Para ter mais informações, consulte [Gerencie suas frotas do Amazon GameLift](#). É possível atualizar a capacidade da frota depois que ela atingir o status ACTIVE. Para ter mais informações, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#). Também é possível adicionar ou remover locais remotos.

Quando terminar de revisar, escolha Criar.

Se sua solicitação for bem-sucedida, o console exibirá a página de detalhes do novo recurso da frota. Inicialmente NEW, o status é: a Amazon GameLift inicia o processo de criação da frota. Você pode rastrear o status da nova frota na página Frotas. Uma frota está pronta para sediar sessões de jogo quando atinge o status ACTIVE.

AWS CLI

Para criar uma frota de contêineres com o AWS CLI, abra uma janela de linha de comando e use o `create-fleet` comando. Para obter mais informações sobre esse comando, consulte [create-fleet](#) na Referência de AWS CLI Comandos.

O exemplo de `create-fleet` solicitação mostrado abaixo cria uma nova frota de contêineres com as seguintes características:

- `ContainerGroupsConfiguration` Especifica uma única definição de grupo de contêineres de réplica: `MegaFrogRaceServer.NA.v2`. Três cópias do grupo de réplicas serão implantadas em cada instância da frota. Cada instância tem 30 portas de conexão disponíveis para acesso aos processos na instância.
- A frota usa instâncias sob demanda `c5.large`.
- Ele implanta grupos de contêineres nos seguintes locais:
 - `us-west-2` (região de origem)
 - `ca-central-1` (localização remota)
- Cada grupo de contêineres de réplica em uma instância executará 5 processos de servidor de jogos simultaneamente, permitindo que cada instância hospede até 15 sessões de jogo por vez.
- Em cada grupo de contêineres de réplica, a Amazon GameLift permite que duas novas sessões de jogo sejam ativadas ao mesmo tempo. Ele também encerra qualquer sessão de jogos ativada se estiverem prontos para hospedar jogadores em um prazo de 300 segundos.
- Todas as sessões de jogos hospedadas em instâncias nesta frota apresentam proteção de sessão de jogos ativada.
- Os jogadores individuais podem criar três novas sessões de jogos dentro de um período de 15 minutos.

```
aws gamelift create-fleet \  
  --name SampleFleet123 \  
  --description "The sample test fleet" \  
  --compute-type "CONTAINER" \  
  --container-groups-configuration  
  "ContainerGroupDefinitionNames=['MegaFrogRaceServer.NA.v2'],  
  DesiredReplicaContainerGroupPerInstance=3,  
  ConnectionPortRange={FromPort=1010,ToPort=1040}" \  
  --ec2-instance-type c5.large \  
  --region us-west-2 \  
  --locations "Location=ca-central-1" \  
  --fleet-type ON_DEMAND \  
  --runtime-configuration "GameSessionActivationTimeoutSeconds=300,  
  MaxConcurrentGameSessionActivations=2, ServerProcesses=[{LaunchPath=/local/game/  
  MegaFrogRace/server.exe,ConcurrentExecutions=5}]" \  

```



```
--new-game-session-protection-policy "FullProtection" \  
--resource-creation-limit-policy "NewGameSessionsPerCreator=3,  
PolicyPeriodInMinutes=15" \  
--ec2-inbound-permissions  
"FromPort=1010,ToPort=1040,IpRange=0.0.0.0/0,Protocol=UDP" \  

```

Se a solicitação de criação de frota for bem-sucedida, a Amazon GameLift retornará um conjunto de atributos da frota que inclui as configurações solicitadas e um novo ID de frota. A Amazon GameLift então define o status da frota e o status da localização como Novo e inicia o processo de ativação da frota. Você pode acompanhar o status da frota e visualizar outras informações da frota usando estes comandos da CLI:

- [describe-fleet-events](#)
- [describe-fleet-attributes](#)
- [describe-fleet-capacity](#)
- [describe-fleet-port-settings](#)
- [describe-fleet-utilization](#)
- [describe-runtime-configuration](#)
- [describe-fleet-location-attributes](#)
- [describe-fleet-location-capacity](#)
- [describe-fleet-location-utilization](#)

É possível alterar a capacidade da frota e outras definições de configuração, conforme necessário, usando estes comandos:

- [update-fleet-attributes](#)
- [update-fleet-capacity](#)
- [update-fleet-port-settings](#)
- [update-runtime-configuration](#)
- [create-fleet-locations](#)
- [delete-fleet-locations](#)

Gerencie suas frotas de GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Quando quiser obter informações sobre sua frota de contêineres ou fazer alterações, você pode usar as seguintes ações para gerenciar sua frota de contêineres.

Visualizar recursos da

Veja a seguir algumas maneiras pelas quais você pode obter informações sobre os recursos em sua frota de contêineres.

- [DescribeCompute](#)- Retorna detalhes sobre um contêiner registrado como computação.
- [DescribeContainerGroupDefinition](#)- Retorna detalhes sobre a definição de um grupo de contêineres. Esse recurso descreve como o grupo e seus contêineres são configurados.
- [DescribeFleetAttributes](#)- Obtém os atributos da frota, que incluem o intervalo da porta de conexão e outros atributos.
- [DescribeFleetCapacity](#)- Obtém uma contagem dos grupos de contêineres de réplicas na frota e seus status.
- [DescribeRuntimeConfiguration](#)- Descreve os processos do servidor que são executados em cada grupo de contêineres de réplica.
- [GetComputeAccess](#)- Fornece acesso remoto a uma instância que hospeda o grupo de contêineres.
- [GetComputeAuthToken](#)- Solicita um token de autenticação da Amazon GameLift para um recurso computacional em uma frota de contêineres.
- [ListCompute](#)- Lista grupos de contêineres registrados como computadores.
- [ListContainerGroupDefinitions](#)- Lista as definições do grupo de contêineres.
- [ListFleets](#)- Lista as frotas que estão usando um grupo específico de contêineres.

Atualizar recursos

Veja a seguir algumas maneiras de criar e modificar recursos da frota de contêineres.

- [CreateContainerGroupDefinition](#)- Cria uma definição de grupo de contêineres.
- [CreateFleet](#)- Cria uma frota de contêineres quando ComputeType está configurado para CONTAINER.

- [RegisterCompute](#)- Registra computadores com uma frota de contêineres.
- [UpdateFleetAttributes](#)- Atualiza os atributos mutáveis de uma frota, como as opções de configuração da frota Anywhere.
- [UpdateFleetCapacity](#)- Atualiza as configurações de capacidade para uma frota EC2 gerenciada ou frota de contêineres.
- [UpdateRuntimeConfiguration](#)- Atualiza a configuração do tempo de execução, que descreve quais processos do servidor devem ser executados em cada grupo de contêineres de réplica registrado como computação.

Excluir recursos

Veja a seguir algumas maneiras pelas quais você pode remover recursos da frota de contêineres.

- [DeleteContainerGroupDefinition](#)- Exclui uma definição de grupo de contêineres.
- [DeleteFleet](#)- Exclui uma frota.
- [DeregisterCompute](#)- Remove um recurso computacional de uma frota de contêineres.

Escalando frotas de GameLift contêineres da Amazon

Esta documentação é para um recurso que está em versão prévia pública. Está sujeita a alteração.

Uma das tarefas mais desafiadoras da hospedagem de jogos é escalar a capacidade para atender à demanda dos jogadores sem desperdiçar dinheiro com recursos que não sejam necessários. Em uma frota de contêineres, você escala a capacidade da sua frota adicionando ou removendo instâncias da frota.

Quando você cria uma nova frota, a Amazon GameLift define a capacidade desejada da frota em uma instância e implanta uma instância na região de origem da frota. Para uma frota com vários locais, a Amazon GameLift implanta uma instância na região de origem e em cada local remoto. Depois que o status da frota ACTIVE for atingido, você poderá aumentar a capacidade desejada para aumentar ou diminuir a capacidade desejada para reduzir a escala.

Você pode usar os recursos de GameLift escalabilidade da Amazon para alterar a capacidade manualmente ou configurar a escalabilidade automática com base na demanda dos jogadores:

- Configure o escalonamento automático com rastreamento de alvos. Consulte [Ajuste de escala automático baseado em destino](#).

- Altere manualmente a capacidade da sua frota. Consulte [Defina manualmente a capacidade de uma frota do Amazon GameLift](#).

Ao escalar uma frota de contêineres, considere como a adição ou remoção de instâncias afeta a capacidade da frota de hospedar sessões de jogos e jogadores.

- Sessões de jogo por instância
 - Cada processo do servidor de jogos em execução em uma instância representa a capacidade de hospedar uma sessão de jogo.
 - Use essa fórmula para calcular o número de sessões de jogo que são executadas simultaneamente em uma instância de frota de contêineres:

```
[Game sessions per instance] = [# of processes per replica container group] * [# of replica container groups per instance]
```

- Para processos por grupo de contêineres de réplica, chame [DescribeRuntimeConfiguration](#) conte o número de execuções simultâneas dos processos do servidor de jogos.
 - Para grupos de contêineres de réplica por instância, ligue [DescribeFleetAttributes](#) para obter o `DesiredReplicaContainerGroupPerInstance` valor. Se esse valor não estiver definido, use o `MaxReplicaContainerGroupsPerInstance` valor.
- Jogadores por instância
 - Você decide o número de vagas de jogador a serem permitidas em cada sessão de jogo. Dependendo de como sua solução de hospedagem lida com o posicionamento da sessão de jogo, você pode definir jogadores por sessão de jogo em sua configuração de matchmaking ou em suas chamadas para iniciar a colocação de uma sessão de jogo.
 - Use essa fórmula para calcular o número de jogadores que podem jogar seu jogo simultaneamente em uma instância de frota de contêineres:

```
[Players per instance] = [# of game sessions per instance] * [# of player slots per game session]
```

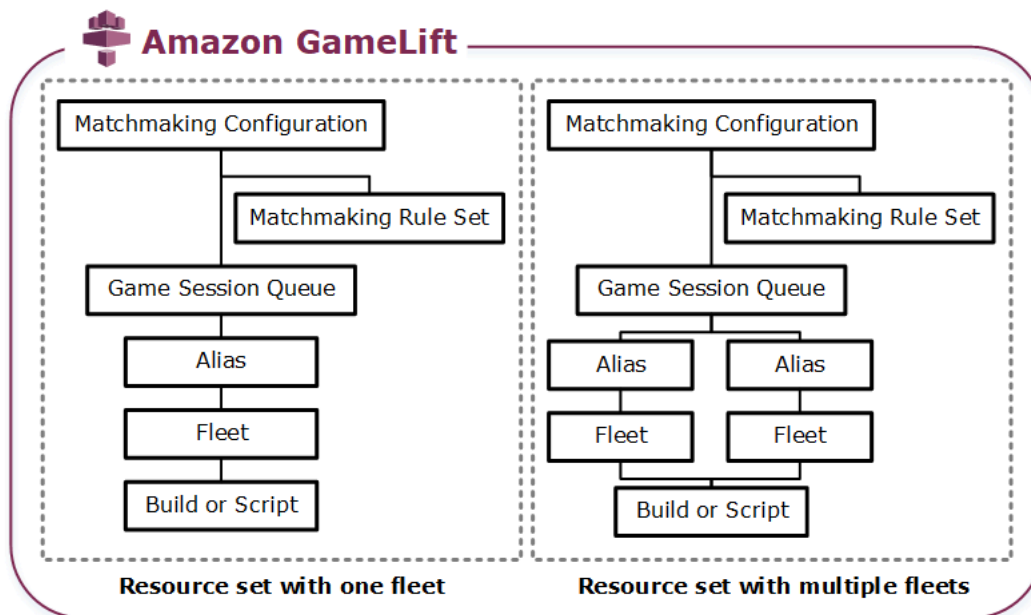
Para obter a capacidade total atual de uma frota de contêineres, ligue para [DescribeFleetCapacity](#) nossa [DescribeFleetLocation Capacidade](#) para obter o número de grupos de contêineres de réplicas na frota. Grupos ativos são aqueles que atualmente hospedam sessões de

jogos. Grupos ociosos estão prontos para sediar uma nova sessão de jogo. Multiplique esses valores pelo número de processos do servidor por grupo de contêineres de réplica.

Gerenciar recursos de hospedagem do Amazon GameLift

Esta seção fornece informações detalhadas sobre como configurar os recursos gerenciados do Amazon GameLift para executar os servidores de jogos e hospedar sessões de jogos para jogadores. É preciso configurar e implantar recursos, escalar a capacidade para atender à demanda dos jogadores e localizar os recursos disponíveis para hospedar sessões de jogo.

O diagrama a seguir ilustra como os objetos de recursos do Amazon GameLift se relacionam entre si. Use uma construção ou script para criar uma frota, atribuir um alias a uma frota e adicionar frotas a uma fila de sessões de jogo usando seu alias. Para jogos que usam a criação de jogos do FlexMatch, use a fila da sessão de jogo e um conjunto de regras de criação de jogos para criar uma configuração de criação de jogos.



Código do servidor de jogos

- **Compilação:** o software de servidor de jogos compilado de forma personalizada, que é executado no Amazon GameLift e hospeda sessões de jogos para seus jogadores. Uma compilação de jogo representa o conjunto de arquivos que executam o servidor de jogos em um determinado sistema operacional e que devem ser integrados ao Amazon GameLift. Carregue os arquivos de compilação de jogo no Amazon GameLift, nas Regiões da AWS, onde você planeja configurar frotas. Para obter mais informações, consulte [Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift](#).
- **Script:** a configuração e a lógica de jogo personalizada para uso com servidores em tempo real. Configure servidores em tempo real para os clientes de jogos criando um script com o uso do

JavaScript e adicione lógica de jogo personalizada para hospedar sessões de jogo para os jogadores. Para obter mais informações, consulte [Carregar um script de servidores em tempo real no Amazon GameLift](#).

Frota

Uma coleção de recursos de computação que executa os servidores de jogos e hospeda sessões de jogos para os jogadores. Para obter informações sobre onde é possível implantar frotas, consulte [Locais GameLift de hospedagem da Amazon](#). Para obter informações sobre como criar frotas, consulte [Configurar as frotas do Amazon GameLift](#).

Alias

Um identificador abstrato de uma frota que você pode usar para alterar a frota à qual os jogadores estão conectados a qualquer momento. Para obter mais informações, consulte [Adicionar um alias a uma frota da Amazon GameLift](#).

Fila de sessões de jogo

Um mecanismo de posicionamento de sessões de jogo que recebe solicitações de novas sessões de jogo e procura servidores de jogos disponíveis para hospedar as novas sessões. Para obter mais informações sobre filas de sessões de jogo e como elas funcionam, consulte [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).

Fazer upload de compilações e scripts para o Amazon GameLift

Antes de implantar servidores de jogos multijogador para hospedagem com o Amazon GameLift, é necessário fazer upload dos arquivos do servidor de jogos. Os tópicos desta seção fornecem orientações sobre como preparar e fazer upload de arquivos de compilação do servidor de jogos personalizado ou arquivos de script do servidor dos servidores em tempo real.

Tópicos

- [Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift](#)
- [Carregar um script de servidores em tempo real no Amazon GameLift](#)

Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift

Depois de integrar seu servidor de jogos com o Amazon GameLift, carregue os arquivos de compilação no Amazon GameLift. Este tópico aborda como empacotar os arquivos de compilação de jogo, criar um script de instalação de compilação opcional e, depois, fazer upload dos arquivos usando a [AWS Command Line Interface \(AWS CLI\)](#) ou um SDK da AWS.

Tópicos

- [Criar um pacote dos arquivos de compilação do jogo](#)
- [Crie uma versão do Amazon GameLift](#)
- [Atualizar os arquivos de compilação](#)
- [Adicionar um script de instalação da compilação](#)

Criar um pacote dos arquivos de compilação do jogo

Antes de fazer o upload do seu servidor de jogo configurado para o Amazon GameLift, empacote os arquivos de compilação do jogo em um diretório de compilação. Esse diretório precisa incluir todos os componentes necessários para executar os servidores de jogos e hospedar as sessões de jogos, inclusive o seguinte:

- Binários do servidor de jogo – Os arquivos binários necessários para executar o servidor de jogo. Uma compilação pode incluir binários para vários servidores de jogos compilados para rodar na mesma plataforma. Para obter uma lista das plataformas com suporte, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).
- Dependências – Qualquer arquivo dependente exigido pelos executáveis do servidor de jogo para ser executado. Entre os exemplos estão ativos, arquivos de configuração e bibliotecas dependentes.

Note

Para compilações de jogos criadas com o SDK do servidor Amazon GameLift para C++ (incluindo aquelas criadas com o plug-in Unreal), inclua a DLL do OpenSSL para a mesma versão do OpenSSL com a qual você criou o SDK do servidor. Consulte o arquivo README em SDK em servidor para obter mais detalhes.

- Script de instalação (opcional) – Um arquivo de script para lidar com tarefas que instalam sua compilação de jogos nos servidores de hospedagem do Amazon GameLift. Coloque este arquivo na raiz do diretório de compilação. O Amazon GameLift executa o script de instalação como parte da criação da frota.

Configure qualquer aplicativo em sua compilação, inclusive o script de instalação, para acessar seus recursos com segurança em outros serviços da AWS. Para obter informações sobre formas de fazer isso, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Depois de empacotar os arquivos de compilação, certifique-se de que o servidor do jogo possa ser executado em uma instalação limpa do sistema operacional de destino. Isso verifica se você incluiu de todas as dependências necessárias no pacote e a precisão do script de instalação.

Crie uma versão do Amazon GameLift

Ao criar uma compilação e fazer upload dos arquivos, você tem algumas opções:

- [Criar uma compilação usando um diretório de arquivos](#). Esse é a opção mais simples e a mais usada.
- [Crie uma compilação com arquivos no Amazon Simple Storage Service \(Amazon S3\)](#). Com essa opção, é possível gerenciar as versões de compilação no Amazon S3.

Com ambos os métodos, o Amazon GameLift cria um recurso de compilação com um ID de compilação exclusivo e outros metadados. A compilação começa no status Inicializado. Após o Amazon GameLift adquirir os arquivos do servidor do jogo, a compilação passa para o status Pronto.

Quando a compilação estiver pronta, você poderá implantá-la em uma nova frota do Amazon GameLift. Para obter mais informações, consulte [Criar uma frota gerenciada pelo Amazon GameLift](#). Quando o Amazon GameLift configura a nova frota, ele baixa os arquivos de construção para cada instância da frota e instala os arquivos de construção.

Criar uma compilação usando um diretório de arquivos

Para criar uma compilação de jogo armazenada em qualquer local, inclusive um diretório local, use o comando [upload-build](#) da AWS CLI. Esse comando cria um registro de compilação no Amazon GameLift e faz upload de arquivos de um local especificado por você.

Envie uma solicitação de upload. Em uma janela da linha de comando, insira o comando `upload-build` e os parâmetros a seguir.

```
aws gamelift upload-build \  
  --name user-defined name of build \  
  --operating-system supported OS \  
  --server-sdk-version Amazon GameLift server SDK version \  
  --build-root build path \  
  --build-version user-defined build number \  
  --region region name
```

- `operating-system` – O ambiente de execução da construção do servidor de jogos. Especifique um valor de OS. Não é possível atualizá-lo mais tarde.
- `server-sdk-version` – A versão do SDK do servidor do Amazon GameLift à qual o servidor de jogos está integrado. Se você não fornecer o valor, o Amazon GameLift usará o valor padrão 4.0.2. Caso especifique uma versão incorreta do SDK do servidor, a compilação do servidor do jogo poderá falhar ao chamar `InitSdk` para estabelecer uma conexão com o serviço do Amazon GameLift.
- `build-root` – O caminho do diretório dos arquivos de compilação.
- `name` – Um nome descritivo para a nova compilação.
- `build-version` – Os detalhes da versão dos arquivos de compilação.
- `region` – A Região AWS onde você deseja criar sua compilação. Crie a compilação na Região onde você planeja implantar frotas. Se você estiver implantando o jogo em várias Regiões, crie uma compilação em cada região.

Note

Visualize sua região padrão atual usando o [aws configure get region](#). Para alterar sua Região padrão, use o comando [aws configure set region *region name*](#).

Exemplos

```
aws gamelift upload-build \  
  --operating-system AMAZON_LINUX_2023 \  
  
  --server-sdk-version "5.0.0" \  
  --build-root "~/mygame" \  
  --name "My Game Nightly Build" \  
  --build-version "build 255" \  
  --region us-east-1
```

```
--region us-west-2
```

```
aws gamelift upload-build \  
  --operating-system WINDOWS_2016 \  
  --server-sdk-version "5.0.0" \  
  --build-root "C:\mygame" \  
  --name "My Game Nightly Build" \  
  --build-version "build 255" \  
  --region us-west-2
```

Em resposta à sua solicitação de upload, o Amazon GameLift fornece o progresso do upload. Em um upload bem-sucedido, o Amazon GameLift retorna o novo ID de registro de compilação. O tempo de upload depende do tamanho dos arquivos de jogo e da velocidade da conexão.

Criar uma compilação com arquivos no Amazon S3

É possível armazenar os arquivos de compilação no Amazon S3 e enviá-los para o Amazon GameLift a partir daí. Ao criar sua compilação, você especifica o local do bucket do S3 e o Amazon GameLift recupera os arquivos de compilação diretamente do Amazon S3.

Para criar um recurso de compilação

1. Armazene os arquivos de compilação no Amazon S3. Crie um arquivo .zip contendo os arquivos de compilação empacotados e faça upload deles para um bucket do S3 na sua Conta da AWS. Anote o rótulo do bucket e o nome do arquivo, você precisará deles ao criar uma compilação do Amazon GameLift.
2. Conceda ao Amazon GameLift acesso aos seus arquivos de compilação. Crie um perfil do IAM seguindo as instruções em [Acesse um arquivo de compilação do jogo no Amazon S3](#). Após ter criado a função, anote o nome do recurso da Amazon (ARN) da nova função, você precisará disso ao criar uma compilação.
3. Criar uma compilação. Use o console do Amazon GameLift ou a AWS CLI o para criar um registro de compilação. Você deve ter a PassRole permissão, conforme descrito em [Exemplos de permissões do IAM do Amazon GameLift](#).

Console

1. No [console do Amazon GameLift](#), no painel de navegação, escolha Hospedagem, Compilações.

2. Na página **Compilações**, escolha **Criar compilação**.
3. Na página **Criar compilação**, em **Configurações de compilação**, faça o seguinte:
 - a. Em **Nome**, insira um nome de script.
 - b. Em **Versão**, insira uma versão. Como você pode atualizar o conteúdo de uma compilação, os dados da versão podem ajudá-lo a monitorar as atualizações.
 - c. Em **SO (Sistema operacional)**, escolha o sistema operacional da versão do seu servidor de jogos. Não é possível atualizar esse valor posteriormente.
 - d. Em **Compilação de servidor de jogo**, insira o URI do S3 do objeto de compilação que você carregou no Amazon S3 e escolha a versão do objeto. Caso não se lembre do URI e da versão do objeto do Amazon S3, escolha **Browse S3** e pesquise o objeto de construção.
 - e. Para o perfil do IAM, escolha a função criada para dar ao Amazon GameLift acesso ao seu bucket do S3 e ao objeto de compilação.
4. (Opcional) Em **Tags**, adicione tags à compilação inserindo pares de **Chave e Valor**.
5. Escolha **Criar**.

O Amazon GameLift atribui uma ID ao nova compilação e carrega o arquivo .zip designado. Visualize a nova compilação, incluindo o status, na página **Compilações**.

AWS CLI

Para definir a nova compilação e fazer upload dos arquivos de compilação do servidor, use o comando [create-build](#).

1. Abra uma janela de linha de comando e alterne para um diretório em que você possa usar a AWS CLI.
2. Digite o comando create-build:

```
aws gamelift create-build \  
  --name user-defined name of build \  
  --server-sdk-version Amazon GameLift server SDK version \  
  --operating-system supported OS \  
  --build-version user-defined build number \  
  --storage-location "Bucket"=S3 bucket label,"Key"=Build .zip file \  
  name,"RoleArn"=Access role ARN} \  
  --region region name
```



```
--region us-west-2
```

3. Para visualizar a nova compilação, use o comando [describe-build](#).

Atualizar os arquivos de compilação

Atualize os metadados de um recurso de compilação usando o console do Amazon GameLift ou o comando [update-build](#) AWS CLI.

Após criar uma compilação do Amazon GameLift, não será possível atualizar os arquivos de compilação associados a ela. Para cada novo conjunto de arquivos, crie uma nova versão do Amazon GameLift. Usando o comando [upload-build](#), o Amazon GameLift cria automaticamente um novo registro de compilação para cada solicitação. Caso forneça arquivos de compilação usando o comando [create-build](#), faça upload de um novo arquivo .zip da compilação com um nome diferente para o Amazon S3 e crie uma compilação referenciando o novo nome de arquivo.

Experimente essas dicas para implantar compilações atualizadas:

- Use filas e frotas de troca conforme necessário. Ao configurar o cliente do jogo com o Amazon GameLift, especifique uma fila em vez de uma frota. Com as filas, adicione as novas frotas com a nova compilação à sua fila e remova as frotas antigas. Para obter mais informações, consulte [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).
- Use aliases a fim de transferir jogadores para uma nova compilação de jogo. Ao integrar o cliente do jogo ao Amazon GameLift, especifique o alias de uma frota, em vez do ID de uma frota. Para obter mais informações, consulte [Adicionar um alias a uma frota da Amazon GameLift](#).
- Configure atualizações de compilação automatizadas. Para exemplos de scripts e informações sobre a incorporação de implantações do Amazon GameLift em seu sistema de compilação, consulte [Automatizar implantações no Amazon GameLift](#) no Blog de tecnologia de jogos da AWS.

Adicionar um script de instalação da compilação

Crie um script de instalação para o OS (sistema operacional) da compilação do seu jogo:

- Windows: crie um arquivo em lote chamado `install.bat`.
- Linux: crie um arquivo de script de shell chamado `install.sh`.

Ao criar um script de instalação, lembre-se do seguinte:

- O script não pode receber nenhuma entrada do usuário.
- O Amazon GameLift instala a compilação e recria os diretórios de arquivos no pacote de compilação em um servidor de hospedagem nos seguintes locais:
 - Frotas do Windows: C:\game
 - Frotas do Linux: /local/game
- Durante o processo de instalação de frotas do Linux, o usuário de execução tem acesso limitado à estrutura de arquivos da instância. Este usuário tem direitos totais para o diretório em que seus arquivos de compilação estão instalados. Se seu script de instalação realizar ações que exijam permissões de administrador, especifique o acesso de administrador usando sudo. O usuário de execução para frotas do Windows tem permissões de administrador por padrão. As falhas de permissão relacionadas ao script de instalação geram uma mensagem de evento que indica um problema com o script.
- No Linux, o Amazon GameLift dá suporte às linguagens comuns de interpretadores de shell, como o bash. Adicione um shebang (como #!/bin/bash) à parte superior do seu script de instalação. Para verificar o suporte para os seus comandos shell preferidos, acesse remotamente uma instância ativa do Linux e abra um prompt de shell. Para obter mais informações, consulte [Conecte-se remotamente às instâncias da GameLift frota da Amazon](#).
- O script de instalação não pode depender de uma conexão de emparelhamento da VPC. Uma conexão de emparelhamento da VPC não está disponível até que o Amazon GameLift instale a compilação em instâncias de frota.

Example Arquivo bash de instalação em Windows

Este exemplo de arquivo `install.bat` instala os componentes de tempo de execução do Visual C++ necessários para que o servidor do jogo e grave os resultados em um arquivo de log. O script inclui o arquivo do componente no pacote de compilação na raiz.

```
vcxbuild.exe /install /quiet /norestart /log c:\game\vcxbuild_2013_x64.log
```

Example Script de shell de instalação do Linux

Esse arquivo `install.sh` de exemplo usa bash no script de instalação e grava os resultados em um arquivo de log.

```
#!/bin/bash  
echo 'Hello World' > install.log
```

Este arquivo `install.sh` de exemplo mostra como você poderá usar o atendente do Amazon CloudWatch para coletar métricas personalizadas e em nível de sistema e executar com a rotação de logs. Como o Amazon GameLift é executado em um serviço de VPC, conceda permissões ao Amazon GameLift para assumir um perfil do IAM do AWS Identity and Access Management em seu nome. Para permitir que o Amazon GameLift assuma uma função, crie uma função que inclua a AWS para política gerenciada `CloudWatchAgentAdminPolicy` e use essa função ao criar uma frota.

```
sudo yum install -y amazon-cloudwatch-agent
sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
sudo yum install -y collectd
cat <<'EOF' > /tmp/config.json
{
  "agent": {
    "metrics_collection_interval": 60,
    "run_as_user": "root",
    "credentials": {
      "role_arn": "arn:aws:iam::account#:role/rolename"
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/tmp/log",
            "log_group_name": "gllog",
            "log_stream_name": "{instance_id}"
          }
        ]
      }
    }
  },
  "metrics": {
    "namespace": "GL_Metric",
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "metrics_collected": {
      // Configure metrics you want to collect.

```



```
        // For more information, see Manually create or edit the CloudWatch agent configuration file.
    }
}
EOF
sudo mv /tmp/config.json /opt/aws/amazon-cloudwatch-agent/bin/config.json
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json
sudo systemctl enable amazon-cloudwatch-agent.service
```

Carregar um script de servidores em tempo real no Amazon GameLift

Quando você estiver pronto para implantar servidores em tempo real no jogo, carregue arquivos concluídos de script de servidores em tempo real no Amazon GameLift. Faça isso criando um recurso de script do Amazon GameLift e especificando a localização dos arquivos de script. Também é possível atualizar os arquivos de script de servidores que já estiverem implantados carregando novos arquivos para um recurso de script existente.

Quando você cria um novo recurso de script, o Amazon GameLift atribui a ele um ID de script exclusivo (por exemplo, `script-1111aaaa-22bb-33cc-44dd-5555eeee66ff`) e carrega uma cópia dos arquivos de script. O tempo de upload depende do tamanho de seus arquivos de script e da velocidade da sua conexão.

Depois de criar o recurso de script, o Amazon GameLift implanta o script com uma nova frota de servidores em tempo real. O Amazon GameLift instala o script de servidores em cada instância da frota, posicionando os arquivos de script em `/local/game`.

Para solucionar problemas de ativação da frota relacionados ao script de servidores, consulte [Depurar problemas com a frota do Amazon GameLift](#).

Empacotar os arquivos de script

Seu script de servidores pode incluir um ou mais arquivos combinados em um único arquivo `.zip` para upload. O arquivo `.zip` deve conter todos os arquivos que o script precisa para ser executado.

É possível armazenar os arquivos de script compactados em um diretório de arquivos local ou em um bucket do Amazon Simple Storage Service (Amazon S3).

Carregar arquivos de script em um diretório local

Se você tiver seus arquivos de script armazenados localmente, é possível enviá-los para o Amazon GameLift a partir de lá. Para criar o recurso de script, use o console Amazon GameLift ou a [AWS Command Line Interface \(AWS CLI\)](#).

Amazon GameLift console

Para criar um recurso de script

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, escolha Hospedagem, Scripts.
3. Na página Scripts, escolha Criar script.
4. Na página Criar script, em Configurações do script, faça o seguinte:
 - a. Para Nome, insira um novo nome.
 - b. (Opcional) Para Versão, insira informações da versão. Como o conteúdo de um script pode ser atualizado, os dados da versão podem ser úteis para rastrear atualizações.
 - c. Para Origem do script, escolha Carregar um arquivo .zip.
 - d. Para Arquivos de script, escolha Escolher arquivo, procure o arquivo .zip que contém o script e, em seguida, escolha esse arquivo.
5. (Opcional) Em Tags, adicione tags ao script inserindo pares de Chave e Valor.
6. Escolha Create (Criar).

O Amazon GameLift atribui uma ID ao novo script e carrega o arquivo .zip designado. É possível visualizar o novo script, incluindo seu status, na página Scripts.

AWS CLI

Use o comando [create-script](#) da AWS CLI para definir o novo script e carregar os arquivos de script de servidores.

Para criar um recurso de script

1. Coloque o arquivo .zip em um diretório onde você possa usar a AWS CLI.
2. Abra uma janela de linha de comando e alterne para o diretório onde você colocou o arquivo .zip.

3. Insira os seguintes parâmetros e o comando `create-script`. Para o parâmetro `--zip-file`, certifique-se de adicionar a string `fileb://` ao nome do arquivo `.zip`. Ela identifica o arquivo como binário para que o Amazon GameLift processe o conteúdo compactado.

```
aws gamelift create-script \  
  --name user-defined name of script \  
  --script-version user-defined version info \  
  --zip-file fileb://name of zip file \  
  --region region name
```

Exemplo

```
aws gamelift create-script \  
  --name "My_Realtime_Server_Script_1" \  
  --script-version "1.0.0" \  
  --zip-file fileb://myrealtime_script_1.0.0.zip \  
  --region us-west-2
```

Em resposta à sua solicitação, o Amazon GameLift retorna o objeto do novo script.

4. Para ver o novo script, chame [describe-script](#).

Carregue arquivos de script no Amazon S3

É possível armazenar os arquivos de script em um bucket do Amazon S3 e carregá-los no Amazon GameLift a partir de lá. Ao criar o script, especifique o local do bucket do S3 e o Amazon GameLift recupera os arquivos de script do Amazon S3.

Para criar um recurso de script

1. Armazene os arquivos de script em um bucket do S3. Crie um arquivo `.zip` contendo os arquivos de script de servidores e carregue-o em um bucket do S3 em uma Conta da AWS que você controle. Anote o URI do objeto – você precisa dele ao criar um script do Amazon GameLift.

Note

O Amazon GameLift não é compatível com o upload feito de buckets do S3 com nomes que contenham um ponto (`.`).

2. Dê ao Amazon GameLift acesso aos seus arquivos de script. Para criar uma função AWS Identity and Access Management (IAM) que permite que o Amazon GameLift acesse o bucket do S3 que contém o script de servidores, siga as instruções em [Configurar uma função de serviço do IAM para a Amazon GameLift](#). Depois de criar a nova função, anote seu nome, que é necessário ao criar um script.
3. Crie um script. Use o console Amazon GameLift ou a AWS CLI para criar um novo registro de script. Para fazer essa solicitação, você precisa ter a permissão `PassRole` do IAM, conforme descrito em [Exemplos de permissões do IAM do Amazon GameLift](#).

Amazon GameLift console

1. No [console do Amazon GameLift](#), no painel de navegação, escolha Hospedagem, Scripts.
2. Na página Scripts, escolha Criar script.
3. Na página Criar script, em Configurações do script, faça o seguinte:
 - a. Para Nome, insira um nome de script.
 - b. (Opcional) Para Versão, insira informações da versão. Como o conteúdo de um script pode ser atualizado, os dados da versão podem ser úteis para rastrear atualizações.
 - c. Para Origem do script, escolha URI do Amazon S3.
 - d. Insira o URI do S3 do objeto de script que carregado no Amazon S3 e, em seguida, escolha a versão do objeto. Se você não se lembra do URI e da versão do objeto do Amazon S3, escolha Procurar S3 e, em seguida, pesquise o objeto de script.
4. (Opcional) Em Tags, adicione tags ao script inserindo pares de Chave e Valor.
5. Escolha Create (Criar).

O Amazon GameLift atribui uma ID ao novo script e carrega o arquivo .zip designado. É possível visualizar o novo script, incluindo seu status, na página Scripts.

AWS CLI

Use o comando [create-script](#) da AWS CLI para definir o novo script e carregar os arquivos de script de servidores.

1. Abra uma janela de linha de comando e alterne para um diretório em que você possa usar a AWS CLI.

2. Insira os seguintes parâmetros e o comando `create-script`. O parâmetro de `--storage-location` especifica o local do bucket do Amazon S3 dos arquivos de script.

```
aws gamelift create-script \  
  --name [user-defined name of script] \  
  --script-version [user-defined version info] \  
  --storage-location "Bucket"=S3 bucket name,"Key"=name of zip file in S3 bucket,"RoleArn"=Access role ARN \  
  --region region name
```

Exemplo

```
aws gamelift create-script \  
  --name "My_Realtime_Server_Script_1" \  
  --script-version "1.0.0" \  
  --storage-location "Bucket"="gamelift-script","Key"="myrealtime_script_1.0.0.zip","RoleArn"="arn:aws:iam::123456789012:role/S3Access" \  
  --region us-west-2
```

Em resposta à sua solicitação, o Amazon GameLift retorna o objeto do novo script.

3. Para visualizar o novo script, chame [describe-script](#).

Atualizar arquivos de script

É possível atualizar os metadados de um recurso de script usando o console do Amazon GameLift ou o comando [update-script](#) da AWS CLI.

Também é possível atualizar o conteúdo do script para um recurso de script. O Amazon GameLift implanta conteúdo de script em todas as instâncias da frota que usam o recurso de script atualizado. Quando o script atualizado é implantado, as instâncias o usam ao iniciar novas sessões de jogo. As sessões de jogo que já estão em execução no momento da atualização não usam o script atualizado.

Para atualizar arquivos de script

- Para arquivos de script armazenados localmente, para carregar o arquivo script .zip atualizado, use o console do Amazon GameLift ou o comando `update-script`.

- Para arquivos de script armazenados em um bucket do Amazon S3, carregue os arquivos de script atualizados no bucket do S3. O Amazon GameLift verifica periodicamente arquivos de script atualizados e os recupera diretamente do bucket do S3.

Configurar as frotas do Amazon GameLift

Esta seção fornece informações detalhadas sobre como projetar, compilar e manter frotas para uso com o Amazon GameLift. Use as frotas do Amazon GameLift para implantar servidores de jogos personalizados e servidores em tempo real.

Uma frota representa os recursos de hospedagem como um conjunto de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) ou hardware físico. A localização de uma frota determina onde as instâncias ou o hardware são implantados para hospedar sessões de jogo para seus jogadores. O tamanho de uma frota e o número de sessões de jogo e jogadores que ela pode suportar conta com o número de instâncias ou da quantidade de hardware que você fornece. Ajuste as instâncias virtuais manualmente ou usando o ajuste de escala automático.

Vários jogos em produção usam mais de uma frota. Use várias frotas, por exemplo, para ter mais de uma versão do servidor de jogos em execução simultaneamente, para fornecer capacidade de backup para frotas spot ou para criar redundância.

Para saber como criar frotas projetadas para as necessidades do seu jogo, inicie com [Guia de criação de frota do Amazon GameLift](#). Depois que a frota estiver em execução, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#), [Adicionar um alias a uma frota da Amazon GameLift](#) e [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).

Tópicos

- [Guia de criação de frota do Amazon GameLift](#)
- [Crie uma frota do Amazon GameLift](#)
- [Gerencie suas frotas do Amazon GameLift](#)
- [Adicionar um alias a uma frota da Amazon GameLift](#)
- [Depurar problemas com a frota do Amazon GameLift](#)
- [Conecte-se remotamente às instâncias da GameLift frota da Amazon](#)

Guia de criação de frota do Amazon GameLift

Este guia de criação aborda as melhores práticas para criar uma frota de recursos de hospedagem para uso com o Amazon GameLift. Escolha uma combinação de recursos de hospedagem e aprenda a configurá-los para se adequar ao seu jogo.

Tópicos

- [Escolhendo os GameLift recursos computacionais da Amazon](#)
- [Gerencie como o Amazon GameLift lança servidores de jogos](#)
- [Use instâncias spot com a Amazon GameLift](#)

Escolhendo os GameLift recursos computacionais da Amazon

Para implantar seus servidores de jogos e hospedar sessões de jogos para seus jogadores, a Amazon GameLift usa recursos do [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) chamados de instâncias ou seu hardware físico. Ao configurar uma nova frota usando instâncias, decida que tipo de instância você precisa e como executar processos de servidor de jogo nelas. Quando uma frota EC2 gerenciada está ativa e pronta para sediar sessões de jogo, você pode adicionar ou remover instâncias conforme necessário para acomodar a demanda dos jogadores.

Você pode implantar seus servidores de GameLift jogos da Amazon em uma combinação de dois tipos de computação:

- **EC2 gerenciado** – As frotas gerenciadas do EC2 usam instâncias do Amazon EC2 para hospedar seus servidores de jogos. A Amazon GameLift gerencia as instâncias e elimina a carga do gerenciamento de hardware e software da hospedagem de seus jogos.
- **Amazon GameLift Anywhere** — GameLift Anywhere As frotas da Amazon usam sua infraestrutura existente para hospedar servidores de jogos, enquanto a Amazon GameLift gerencia sua organização de partidas e filas.

Ao escolher os recursos de computação para a frota, considere os seguintes fatores:

- [Hardware disponível](#)
- [Localização da frota](#)
- [Instâncias sob demanda versus instâncias spot](#)
- [Sistemas operacionais](#)
- [Tipos de instância](#)

- [Cotas de serviço](#)

Hardware disponível

Considere a infraestrutura existente na implementação. Enquanto você migra jogos para a Amazon GameLift, você pode continuar usando sua infraestrutura. Com a Amazon GameLift Anywhere, você pode usar sua própria infraestrutura junto com as instâncias EC2 GameLift gerenciadas pela Amazon. Você também pode usar sua infraestrutura existente para hospedar jogos mais perto de seus jogadores do que os GameLift locais compatíveis da Amazon podem permitir. Para obter mais informações sobre a configuração de GameLift Anywhere frotas da Amazon, consulte [Crie uma GameLift Anywhere frota da Amazon](#).

Localização da frota

Considere as localizações geográficas em que você planeja implantar seus servidores de jogos. A disponibilidade do tipo de instância varia de Região da AWS acordo com a zona local.

Para frotas com vários locais, a disponibilidade e as cotas das instâncias dependem de uma combinação da região de origem da frota e dos locais remotos selecionados. Para obter mais informações sobre localizações de frotas, consulte [Locais GameLift de hospedagem da Amazon](#).

Para GameLift Anywhere frotas da Amazon, você determina a localização do seu hardware físico. Para obter mais informações sobre localizações personalizadas, consulte [Amazon GameLift Anywhere](#).

Instâncias sob demanda versus instâncias spot

As instâncias sob demanda e as instâncias spot do Amazon EC2 oferecem o mesmo hardware e desempenho, mas diferem em disponibilidade e custo.

Instâncias sob demanda

Você poderá adquirir uma instância sob demanda quando precisar e mantê-la pelo tempo que desejar. As instâncias sob demanda têm um custo fixo, o que significa que você paga pelo tempo que usá-las. Além disso, não há compromissos de longo prazo.

Instâncias spot

As instâncias spot podem oferecer uma alternativa econômica às instâncias sob demanda, utilizando a capacidade de computação não AWS utilizada. Os preços das Instâncias Spot flutuam com base na oferta e na demanda de cada tipo de instância em cada local. AWS pode interromper

as Instâncias Spot sempre que precisar da capacidade de volta. A Amazon GameLift usa filas e o algoritmo FleeTiQ para determinar se AWS isso interromperá uma instância spot, colocando a instância em um estado de reciclagem. Então, quando não há sessões de jogo ativas na instância, a Amazon GameLift tenta substituí-la.

Para obter mais informações sobre como usar as instância spot, consulte [Use instâncias spot com a Amazon GameLift](#).

Sistemas operacionais

GameLift As instâncias da Amazon oferecem suporte a compilações de servidores de jogos que são executadas no Microsoft Windows ou no Amazon Linux. Ao fazer o upload de uma versão de jogo para a Amazon GameLift, especifique o sistema operacional do jogo. Quando você cria uma frota do Amazon EC2 para implantar a compilação do jogo, a Amazon configura GameLift automaticamente as instâncias com o sistema operacional da compilação. Para obter mais informações sobre os sistemas operacionais de servidores de jogos compatíveis, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).

Ao usar uma GameLift Anywhere frota da Amazon, você pode usar qualquer sistema operacional compatível com seu hardware. GameLift AnywhereAs frotas da Amazon exigem que você implante sua versão de jogo no hardware enquanto usa GameLift a Amazon para gerenciar seus recursos em um só lugar.

Tipos de instância

O tipo de instância de uma frota do Amazon EC2 determina o tipo de hardware que as instâncias usam. Diferentes tipos de instância oferecem combinações diferentes de capacidade de computação, memória, armazenamento e recursos de rede.

Ao escolher entre os tipos de instância disponíveis para seu jogo, considere:

- A arquitetura computacional do seu servidor de jogos: x64 ou Arm (AWS Graviton).

Note

As instâncias do Graviton Arm exigem um GameLift servidor Amazon baseado no sistema operacional Linux. O Server SDK 5.1.1 ou mais recente é necessário para C++ e C#. O Server SDK 5.0 ou mais recente é necessário para o Go. Essas instâncias não oferecem out-of-the-box suporte para a instalação do Mono no Amazon Linux 2023 (AL2023) ou no Amazon Linux 2 (AL2).

- Os requisitos de computação, memória e armazenamento da versão do seu servidor de jogos.
- O número de processos do servidor que você planeja executar por instância.

Ao usar um tipo de instância maior, você poderá executar vários processos de servidor em cada instância. Isso pode reduzir o número de instâncias necessárias para atender à demanda dos jogadores.

Para obter mais informações:

- Sobre os tipos de instância, veja [Tipos de instância do Amazon EC2](#).
- Sobre a execução de vários processos por instância, consulte [Gerencie como o Amazon GameLift lança servidores de jogos](#).

Cotas de serviço

Para ver as cotas de serviço padrão para a Amazon GameLift e as cotas atuais para você Conta da AWS, faça o seguinte:

- Para obter informações gerais sobre cotas de serviços para a Amazon GameLift, consulte os [GameLift endpoints e cotas da Amazon](#) no. Referência geral da AWS
- Para obter uma lista dos tipos de instância disponíveis por local para sua conta, abra a página [de cotas de serviços](#) do GameLift console da Amazon. Essa página também exibe o uso atual da sua conta para cada tipo de instância em cada local.
- Para obter uma lista das cotas atuais da sua conta para tipos de instância por região, execute o comando AWS Command Line Interface [describe-ec2-instance-limits](#)(AWS CLI). Esse comando retorna o número de instâncias ativas que você tem na sua região padrão (ou em outra região especificada).

Ao se preparar para lançar seu jogo, preencha um questionário de lançamento no [GameLift console da Amazon](#). A GameLift equipe da Amazon usa o questionário de lançamento para determinar as cotas e limites corretos para o seu jogo.

Gerencie como o Amazon GameLift lança servidores de jogos

Defina uma configuração de runtime de uma frota EC2 gerenciada para executar vários processos de servidor de jogos por instância. Isso usa os recursos de hospedagem com mais eficiência.

Como uma frota gerencia vários processos

O Amazon GameLift usa a configuração de runtime de uma frota para determinar o tipo e o número de processos a serem executados em cada instância. Uma configuração de runtime contém pelo menos uma configuração de processo do servidor que representa um executável do servidor de jogos. É possível definir configurações adicionais do processo do servidor para executar outros tipos de processos relacionados ao seu jogo. Cada configuração do processo do servidor contém as seguintes informações:

- O nome do arquivo e o caminho de um executável na compilação do seu jogo.
- (Opcional) Parâmetros a serem passados para o processo na inicialização
- O número de processos a serem executados simultaneamente.

Quando uma instância na frota é ativada, ela inicia o conjunto de processos do servidor definido na configuração do runtime. Com vários processos, o Amazon GameLift escalona a inicialização de cada processo. Os processos do servidor têm uma vida útil limitada. Ao final, o Amazon GameLift lança novos processos para manter o número e o tipo de processos do servidor definidos na configuração do runtime.

É possível alterar a configuração de runtime a qualquer momento adicionando, alterando ou removendo as configurações de processo do servidor. Cada instância verifica regularmente se há atualizações na configuração de runtime da frota, para implementar as alterações. Veja a seguir como o Amazon GameLift adota as alterações de configuração de runtime:

1. A instância envia uma solicitação ao Amazon GameLift para obter a versão mais recente da configuração de runtime.
2. A instância compara os processos ativos com a configuração de runtime mais recente e, em seguida, faz o seguinte:
 - Se a configuração atualizada do runtime remover um tipo de processo de servidor, os processos de servidor ativos desse tipo continuarão em execução até concluírem. A instância não substitui esses processos do servidor.
 - Se a configuração atualizada do runtime diminuir o número de processos simultâneos para um tipo de processo de servidor, os processos de servidor em excesso desse tipo continuarão em execução até terminarem. A instância não substitui esses processos excessivos do servidor.
 - Se a configuração atualizada do runtime adicionar um novo tipo de processo de servidor ou aumentar os processos simultâneos para um tipo existente, a instância iniciará novos processos

de servidor, até o limite do Amazon GameLift. Nesse caso, a instância inicia novos processos de servidor quando os processos existentes são concluídos.

Otimizar uma frota para vários processos

Para usar vários processos em uma frota, faça o seguinte:

- [Crie uma compilação](#) que contenha os executáveis do servidor de jogos a serem implantados em uma frota e, em seguida, carregue a compilação no Amazon GameLift. Todos os servidores de jogos em uma compilação devem ser executados na mesma plataforma e usar o Server SDK do Amazon GameLift.
- Crie uma configuração de runtime com uma ou mais configurações de processo de servidor e vários processos simultâneos.
- Integre clientes de jogos com a versão do SDK 2016-08-04 da AWS ou posterior.

Para otimizar o desempenho da frota, é recomendável fazer o seguinte:

- Processe cenários de desligamento do processo de servidor para que o Amazon GameLift possa reciclar processos de maneira eficiente. Por exemplo:
 - Adicione um procedimento de encerramento ao código do servidor de jogos que chama a API de servidor de `ProcessEnding()`.
 - Implemente a função de retorno de chamada `OnProcessTerminate()` no código do servidor de jogos para processar as solicitações de encerramento do Amazon GameLift.
- Garanta que o Amazon GameLift desligue e reinicie processos de servidor não íntegros. Relate o status de integridade ao Amazon GameLift implementando a função de retorno de chamada `OnHealthCheck()` no código do servidor do jogo. O Amazon GameLift desliga automaticamente processos de servidor reconhecidamente não íntegros por três relatórios consecutivos. Se você não implementar `OnHealthCheck()`, o Amazon GameLift assumirá que um processo de servidor está íntegro, a menos que o processo não responda a uma comunicação.

Escolha o número de processos por instância

Ao decidir em relação ao número de processos simultâneos a serem executados em uma instância, lembre-se do seguinte:

- O Amazon GameLift limita cada instância a um [número máximo de processos simultâneos](#). A soma de todos os processos simultâneos das configurações de processos do servidor de uma frota não pode exceder essa cota.
- Para manter níveis de desempenho aceitáveis, o tipo de instância do Amazon EC2 pode limitar o número de processos que podem ser executados simultaneamente. Teste diferentes configurações para o jogo a fim de encontrar o número certo de processos para o tipo de instância preferido.
- O Amazon GameLift não executa mais processos simultâneos do que o número total configurado. Isso significa que a transição da configuração de runtime anterior para a nova configuração pode ocorrer gradualmente.

Use instâncias spot com a Amazon GameLift

Ao configurar sua frota EC2 GameLift gerenciada pela Amazon, você pode usar instâncias spot, instâncias sob demanda ou uma combinação. Saiba mais sobre como a Amazon GameLift usa instâncias spot em [Instâncias sob demanda versus instâncias spot](#). Para usar frotas spot, sua integração com o jogo requer os ajustes listados nesta página.

Você está usando FlexMatch para matchmaking? É possível adicionar frotas spot às suas filas de sessões de jogo existentes para colocações de criação de jogos.

1. Crie sua fila de sessões de jogo para instâncias spot.

Gerenciar o posicionamento da sessão de jogo com uma fila é sempre uma melhor prática necessária ao usar instâncias spot. Para projetar a fila, considere o seguinte:

- Locais: para obter a melhor experiência de jogador, escolha locais geograficamente próximos aos seus jogadores.
- Tipos de instância: considere os requisitos de hardware dos servidores de jogos e a disponibilidade de instâncias nos locais que você escolheu.

Para experimentar uma fila que otimize a disponibilidade e a resiliência do spot, consulte [Tutorial: Configurar uma fila de sessão de jogo para instâncias spot](#).

2. Crie as frotas para a sua fila otimizada para spot.

Com base no design da fila, crie frotas para implantar os servidores de jogos nos locais e tipos de instância desejados. Consulte [Criar uma frota gerenciada pelo Amazon GameLift](#) para receber ajuda sobre como criar e configurar novas frotas.

3. Criar uma fila de sessões de jogos.

Adicione os destinos da frota, configure o processo de posicionamento da sessão de jogo e defina as prioridades de posicionamento. Consulte [Criar uma fila de sessões de jogo](#) para receber ajuda sobre como criar e configurar a nova fila.


4. Atualize o serviço de cliente do jogo para usar a fila.

Quando o cliente do jogo usa uma fila para solicitar recursos, a fila evita recursos com alta chance de interrupção e seleciona o local que corresponde às suas prioridades definidas. Para receber ajuda sobre como implantar posicionamentos de sessões de jogo no seu jogo, consulte [Criar sessões de jogos](#).

5. Atualize o servidor de jogos lidar com interrupções spot.

AWS pode interromper as Instâncias Spot com uma notificação de 2 minutos, quando precisar recuperar a capacidade. Configure o servidor de jogos para lidar com interrupções e minimizar o impacto do jogador.

Antes de AWS recuperar uma Instância Spot, ela envia uma notificação de encerramento. A Amazon GameLift passa a notificação para todos os processos afetados do servidor invocando a função de retorno de chamada do Amazon GameLift Server SDK. `onProcessTerminate()` Implemente esse retorno de chamada para encerrar a sessão de jogo ou mover a sessão de jogo e os jogadores para uma nova instância. Consulte [Responder a uma notificação de desligamento do processo do servidor](#) para receber ajuda sobre como implementar `onProcessTerminate()`.

 Note

AWS faz todos os esforços para fornecer a notificação antes de recuperar uma instância, mas é possível que ela AWS recupere a Instância Spot antes que o aviso chegue. Prepare o servidor de jogos para lidar com interrupções inesperadas.

6. Revise o desempenho das frotas spot e das filas.

Veja GameLift as métricas da Amazon no GameLift console da Amazon ou com CloudWatch a Amazon para analisar o desempenho. Para obter mais informações sobre GameLift as métricas da Amazon, consulte [Monitorar o Amazon GameLift com o Amazon CloudWatch](#). As principais métricas incluem:

- Taxa de interrupção: use as métricas InstanceInterruptions e GameSessionInterruptions para acompanhar a quantidade e a frequência das interrupções relacionadas a instâncias spot e sessões de jogos. As sessões de jogo que são recuperadas por AWS têm um status de TERMINATED e um motivo de status de INTERRUPTED
- Eficácia da fila: acompanhe as taxas de sucesso de posicionamento, tempo médio de espera e profundidade da fila, para verificar se o uso de frotas spot não tem impacto no desempenho da fila.
- Uso da frota: monitore dados sobre instâncias, sessões de jogos e sessões de jogadores. O uso de frotas sob demanda pode ser um indicador de que as filas estão evitando posicionamentos em suas frotas spot para evitar interrupções.

Crie uma frota do Amazon GameLift

Crie uma nova frota e implante sua versão personalizada de servidor de jogos ou servidores em tempo real para hospedagem. É possível implantar qualquer recurso de criação de jogo ou script que você envie para o Amazon GameLift.

Tópicos

- [Como funciona a criação de frota do Amazon GameLift](#)
- [Criar uma frota gerenciada pelo Amazon GameLift](#)
- [Crie uma GameLift Anywhere frota da Amazon](#)

Como funciona a criação de frota do Amazon GameLift

Ao criar uma nova frota, o Amazon GameLift inicia um fluxo de trabalho que cria uma frota com uma instância do Amazon Elastic Compute Cloud (Amazon EC2) em cada local da frota. À medida que o Amazon GameLift conclui cada etapa do fluxo de trabalho, a frota emite eventos e o Amazon GameLift atualiza o status da frota. Acompanhe todos os eventos usando o console do Amazon

GameLift ou chamando a operação de [DescribeFleetEvents](#) da API do Amazon GameLift. Também é possível rastrear o status de locais individuais usando [DescribeFleetLocationAttributes](#).

Fluxo de trabalho de criação de frota EC2:

- O Amazon GameLift cria um recurso de frota na região de origem da frota e em cada local remoto definido na frota.
- O Amazon GameLift define a capacidade desejada para uma instância.
- O Amazon GameLift define o status da frota e da localização como Novo.
- O Amazon GameLift começa a gravar eventos no log de eventos da frota.
- O Amazon GameLift aloca os recursos de computação solicitados para uma nova instância em cada localização da frota.
- O Amazon GameLift baixa os arquivos do servidor do jogo para cada instância e define o status da frota como Baixando.
- O Amazon GameLift valida os arquivos baixados do servidor de jogos em cada instância para verificar se nenhum erro ocorreu durante o download. O Amazon GameLift define o status da frota como Validando.
- O Amazon GameLift cria o servidor do jogo em cada instância e define o status da frota como Desenvolvimento.
- O Amazon GameLift começa a iniciar os processos de servidor em cada instância, seguindo as instruções na configuração de runtime da frota. Caso configurou a frota para executar vários processos de servidor simultâneos por instância, o Amazon GameLift escalona a execução do processo em alguns segundos. À medida que cada processo fica on-line, ele reporta a prontidão ao Amazon GameLift. O Amazon GameLift define o status da frota como Ativando.
- O Amazon GameLift define o status da frota e o status da localização como Ativo à medida que os processos do servidor relatam a prontidão.

Amazon GameLift Anywhere fleet creation

- O Amazon GameLift cria um recurso de frota. Para a região de origem da frota e cada local personalizado definido na frota, o Amazon GameLift define o status da frota e do local como Novo.
- O Amazon GameLift começa a gravar eventos no log de eventos da frota.
- Depois que um processo de servidor em uma frota notifica o Amazon GameLift de que está pronto, o Amazon GameLift define o status da frota e o status da localização como Ativo. À medida que

os processos do servidor em outros locais da frota relatam que estão prontos, o Amazon GameLift define o status de cada localização da frota como Ativo.

Para obter ajuda na solução de problemas de criação de frotas, consulte [Depurar problemas com a frota do Amazon GameLift](#).

Criar uma frota gerenciada pelo Amazon GameLift

Use o [console do Amazon GameLift](#) ou o AWS Command Line Interface (AWS CLI) para criar uma frota gerenciada.

Depois de criar uma frota gerenciada do EC2, o status da frota passa por vários estágios conforme o Amazon GameLift implanta a frota, instala e inicia os servidores de jogos. A frota está pronta para hospedar sessões de jogo, após atingir o status ACTIVE. Para obter ajuda com problemas de criação de frotas, consulte [Depurar problemas com a frota do Amazon GameLift](#).

Console

Para criar uma frota EC2 gerenciada

1. No [console do Amazon GameLift](#), no painel de navegação, selecione a opção Frotas.
2. Na página Frotas, selecione Criar frota.
3. Selecione a opção EC2 gerenciado.
4. Na página Detalhes da frota faça o seguinte:
 - a. Para Nome, insira um novo nome de frota. É recomendável incluir o tipo de frota (spot ou sob demanda) no nome da frota. Isso facilita muito a identificação dos tipos de frota ao visualizar uma lista de frotas.
 - b. Em Descrição, forneça uma breve descrição da frota.
 - c. Para o Tipo binário, selecione Compilação ou Script para definir o tipo de servidor de jogo que o Amazon GameLift implanta nessa frota.
 - d. Selecione um script ou compilação na lista suspensa de scripts ou compilações enviadas.
5. (Opcional) Em Detalhes adicionais para o seguinte:
 - a. Para a Função de instância, especifique um perfil do IAM que autorize os aplicativos na compilação de jogo a acessar outros recursos da AWS em sua conta. Para obter

mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#). Para criar uma frota com uma função de instância, sua conta precisa ter a permissão de `PassRole` do IAM. Para obter mais informações, consulte [Exemplos de permissões do IAM do Amazon GameLift](#).

Se você quiser autorizar aplicativos que não sejam executáveis do servidor, como um atendente do CloudWatch, habilite a opção de credenciais compartilhadas.

Não é possível atualizar essas configurações após a criação da frota.

- b. Para a Geração da certificação, escolha que o Amazon GameLift gere um certificado TLS para a frota. Você pode usar um certificado TLS de frota para que o cliente do jogo autentique um servidor de jogos ao se conectar e criptografe todas as comunicações entre cliente e servidor. Para cada instância em uma frota habilitada para TLS, o Amazon GameLift também cria uma nova entrada DNS com o certificado. Use esses recursos para configurar a autenticação e a criptografia para o seu jogo.
- c. Em Grupo de métricas, insira o nome de um grupo de métricas de frota novo ou existente. É possível agregar as métricas de várias frotas, adicionando-as ao mesmo grupo de métricas.

Não é possível atualizar o grupo de métricas após a criação da frota.

6. Selecione a opção Próximo.
7. Na página Selecionar locais, selecione um ou mais locais remotos adicionais para implantar instâncias. A região de origem é selecionada automaticamente com base na região da qual você está acessando o console. Caso selecione locais adicionais, as instâncias da frota também serão implantadas nesses locais.


Important

Para usar regiões que não estão habilitadas por padrão, ative-as na sua Conta da AWS.

- As frotas com regiões não habilitadas que você criou antes de 28 de fevereiro de 2022 não são afetadas.
- Para criar novas frotas com vários locais ou atualizar frotas com vários locais existentes, primeiro ative todas as regiões que você escolher usar.

Para obter mais informações sobre regiões que não estão habilitadas por padrão e como habilitá-las, consulte [Gerenciando Regiões da AWS](#) no Referência geral da AWS.

8. Selecione a opção Próximo.
9. Na página Definir detalhes da instância, escolha
 - a. Instâncias Sob demanda ou Spot para essa frota. Para obter informações sobre tipos de frotas, consulte [Instâncias sob demanda versus instâncias spot](#).
 - b. No menu Arquitetura do filtro, escolha x64 ou Arm.

 Note

As instâncias do Graviton Arm exigem um servidor do Amazon GameLift compilado no sistema operacional do Linux. O Server SDK 5.1.1 ou mais recente é necessário para C++ e C#. O Server SDK 5.0 ou mais recente é necessário para o Go. Essas instâncias não oferecem suporte imediato para a instalação do Mono no Amazon Linux 2023 (AL2023) ou no Amazon Linux 2 (AL2).

Para obter informações sobre as arquiteturas Arm do Amazon EC2, consulte [Tipos de instância do Processador Graviton da AWS](#) e do [Amazon EC2](#).

Para obter informações sobre os tipos de instância com suporte pelo Amazon GameLift, consulte os valores EC2InstanceType nos [parâmetros de solicitação de CreateFleet\(\)](#).

10. Selecione um tipo de instância de Amazon EC2 na lista. Para obter mais informações sobre a escolha de um tipo de instância, consulte [Tipos de instância](#). Após criar a frota, não será possível alterar o tipo de instância.
11. Selecione a opção Próximo.
12. Na página Configurar runtime, em Configuração de runtime, faça o seguinte:
 - a. Em Caminho de execução, insira o caminho do executável do jogo na compilação ou script. Em instâncias do Windows, os servidores de jogos são criados no caminho C : \game. Em instâncias do Linux, os servidores de jogos são criados para /local/game.

Exemplos: **C:\game\MyGame\server.exe**, **/local/game/MyGame/server.exe** ou **MyRealtimeLaunchScript.js**.

- b. (Opcional) Em Parâmetros de inicialização, insira as informações a serem passadas para o executável do jogo como um conjunto de parâmetros da linha de comando. Exemplo: **+sv_port 33435 +start_lobby**.
- c. Em Processos simultâneos, escolha o número de processos do servidor a serem executados simultaneamente em cada instância da frota. Analise os [limites](#) do Amazon GameLift no número de processos simultâneos do servidor.

Limites de processos de servidor simultâneos por instância se aplicam ao total de processos simultâneos para todas as configurações. Se você configurar a frota para exceder o limite, ela não poderá ser ativada.

13. Em Ativação da sessão de jogo, forneça limites para a ativação de novas sessões de jogo nas instâncias dessa frota:
 - a. Para a Ativação máxima da sessão de jogos concorrentes, insira a quantidade de sessões de jogos em uma instância que são ativadas simultaneamente. Este limite é útil quando a execução de várias novas sessões de jogos pode afetar o desempenho de outras sessões de jogos em execução na instância.
 - b. Em Novo tempo limite de ativação, insira quanto tempo esperar pela ativação de uma sessão. Se a sessão do jogo não passar para o status ACTIVE antes do tempo limite, o Amazon GameLift encerra a ativação da sessão do jogo.
14. (Opcional) Em Configurações da porta EC2, faça o seguinte:
 - a. Selecione a opção Adicionar configurações de porta para definir permissões de acesso para tráfego de entrada conectado ao processo do servidor implantado na frota.
 - b. Em Tipo, escolha TCP personalizado ou UDP personalizado.
 - c. Em Intervalo de portas, insira um intervalo de números de portas que permitem conexões de entrada. Um intervalo de portas deve usar o formato `nnnnn[-nnnnn]`, com valores entre 1026 e 60000. Exemplo: **1500** ou **1500-20000**.
 - d. Em Intervalo de endereço IP, insira um intervalo de endereço IP. Use a notação CIDR. Exemplo: **0.0.0.0/0** (Este exemplo permite o acesso de qualquer pessoa que esteja tentando se conectar.)
15. (Opcional) Em Configurações de recursos da sessão de jogo, faça o seguinte:

- a. Para a Política de proteção de escalabilidade de jogos, ative ou desative a proteção de escalabilidade. O Amazon GameLift não encerrará a instância com proteção durante um evento de redução caso esteja hospedando uma sessão de jogo ativa.
 - b. Em Limite de criação de recursos, insira o número máximo de sessões de jogo que um jogador pode criar durante o período da política.
16. Selecione a opção Próximo.
 17. (Opcional) Adicione tags à compilação inserindo pares de Chave e Valor. Selecione a opção Avançar para continuar com a revisão da criação da frota.
 18. Selecione a opção Criar. O Amazon GameLift atribui um ID à nova frota e inicia o processo de ativação da frota. Você pode rastrear o status da nova frota na página Frotas.

Atualize os metadados da frota e a configuração a qualquer momento, independentemente do status da frota. Para obter mais informações, consulte [Gerencie suas frotas do Amazon GameLift](#). É possível atualizar a capacidade da frota depois que ela atingir o status ACTIVE. Para obter mais informações, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#). Também é possível adicionar ou remover locais remotos.

AWS CLI

Para criar uma frota com a AWS CLI, abra uma janela de linha de comando e use o comando `create-fleet`. Para obter mais informações sobre o comando `create-fleet`, consulte [create-fleet](#) na Referência de comandos da AWS CLI.

A solicitação de exemplo `create-fleet` mostrada a seguir cria uma nova frota com as seguintes características:

- A frota usa as instâncias sob demanda `c5.large` com o sistema operacional apropriado para a compilação de jogo selecionada.
- Ele implanta a compilação do servidor de jogos especificada, que deve apresentar um status Pronto nos locais a seguir:
 - `us-west-2` (região de origem)
 - `sa-east-1` (localização remota)
- A geração de certificado TLS está habilitada.
- Cada instância na frota executará simultaneamente dez processos idênticos do servidor de jogos. Dessa forma, cada uma delas poderá hospedar até dez sessões de jogos ao mesmo tempo.

- Em cada instância, o Amazon GameLift permite que duas novas sessões de jogo sejam ativadas simultaneamente. Ele também encerra qualquer sessão de jogos ativada se estiverem prontos para hospedar jogadores em um prazo de 300 segundos.
- Todas as sessões de jogos hospedadas em instâncias nesta frota apresentam proteção de sessão de jogos ativada.
- Os jogadores individuais podem criar três novas sessões de jogos dentro de um período de 15 minutos.
- Cada sessão de jogos hospedada nesta frota tem um ponto de conexão que se enquadra no endereço IP especificado e nos intervalos de portas.
- O Amazon GameLift adiciona métricas dessa frota ao grupo de métricas `EMEAfleets`, que (neste exemplo) combina métricas de todas as frotas nas regiões EMEA.

```
aws gamelift create-fleet \  
  --name SampleFleet123 \  
  --description "The sample test fleet" \  
  --ec2-instance-type c5.large \  
  --region us-west-2 \  
  --locations "Location=sa-east-1" \  
  --fleet-type ON_DEMAND \  
  --build-id build-92f061ed-27c9-4a02-b1f4-6f85b2385620 \  
  --certificate-configuration "CertificateType=GENERATED" \  
  --runtime-configuration "GameSessionActivationTimeoutSeconds=300,  
MaxConcurrentGameSessionActivations=2, ServerProcesses=[{LaunchPath=C:\game  
\Bin64.dedicated\MultiplayerSampleProjectLauncher_Server.exe, Parameters=+sv_port  
33435 +start_lobby, ConcurrentExecutions=10}]" \  
  --new-game-session-protection-policy "FullProtection" \  
  --resource-creation-limit-policy "NewGameSessionsPerCreator=3,  
PolicyPeriodInMinutes=15" \  
  --ec2-inbound-permissions  
"FromPort=33435,ToPort=33435,IpRange=0.0.0.0/0,Protocol=UDP"  
"FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP" \  
  --metric-groups "EMEAfleets"
```

Se a solicitação `create-fleet` for bem-sucedida, o Amazon GameLift retornará um conjunto de atributos da frota dentre os quais estão as definições de configuração solicitadas por você e um novo ID de frota. Em seguida, o Amazon GameLift inicia o processo de ativação da frota e define o status da frota e o status da localização como `Novo`. Você pode acompanhar o status da frota e visualizar outras informações da frota usando estes comandos da CLI:

- [describe-fleet-events](#)
- [describe-fleet-attributes](#)
- [describe-fleet-capacity](#)
- [describe-fleet-port-settings](#)
- [describe-fleet-utilization](#)
- [describe-runtime-configuration](#)
- [describe-fleet-location-attributes](#)
- [describe-fleet-location-capacity](#)
- [describe-fleet-location-utilization](#)

É possível alterar a capacidade da frota e outras definições de configuração, conforme necessário, usando estes comandos:

- [update-fleet-attributes](#)
- [update-fleet-capacity](#)
- [update-fleet-port-settings](#)
- [update-runtime-configuration](#)
- [create-fleet-locations](#)
- [delete-fleet-locations](#)

Crie uma GameLift Anywhere frota da Amazon

Use GameLift a Amazon para integrar o hardware do seu ambiente à sua hospedagem de GameLift jogos na Amazon. A Amazon GameLift Anywhere registra seu hardware GameLift na Amazon em uma Anywhere frota. É possível integrar Anywhere e gerencie frotas EC2 em filas de marcador de jogos e sessões de jogos para gerenciar a marcação de jogos e o posicionamento de jogos.

Para obter mais informações sobre como testar seus servidores de jogos com a Amazon GameLift Anywhere, consulte [Teste sua integração usando frotas Anywhere do Amazon GameLift](#).

Para começar, use a [Suporte ao desenvolvimento com a Amazon GameLift](#) versão 5 ou superior e analise os seguintes conceitos para usar uma GameLift Anywhere frota da Amazon.

Localizações personalizadas

GameLift AnywhereAs frotas da Amazon usam locais personalizados para representar os locais físicos da sua infraestrutura.

Criar uma política personalizada

Para que uma GameLift Anywhere frota da Amazon se comunique com seus recursos computacionais, primeiro registre seu dispositivo. Você pode concluir o registro do dispositivo a partir do Amazon GameLift AWS SDK usando a [RegisterCompute](#) operação. Essa operação usa o endereço IP do dispositivo para associá-lo a uma localização da frota e se comunicar com a Amazon GameLift.

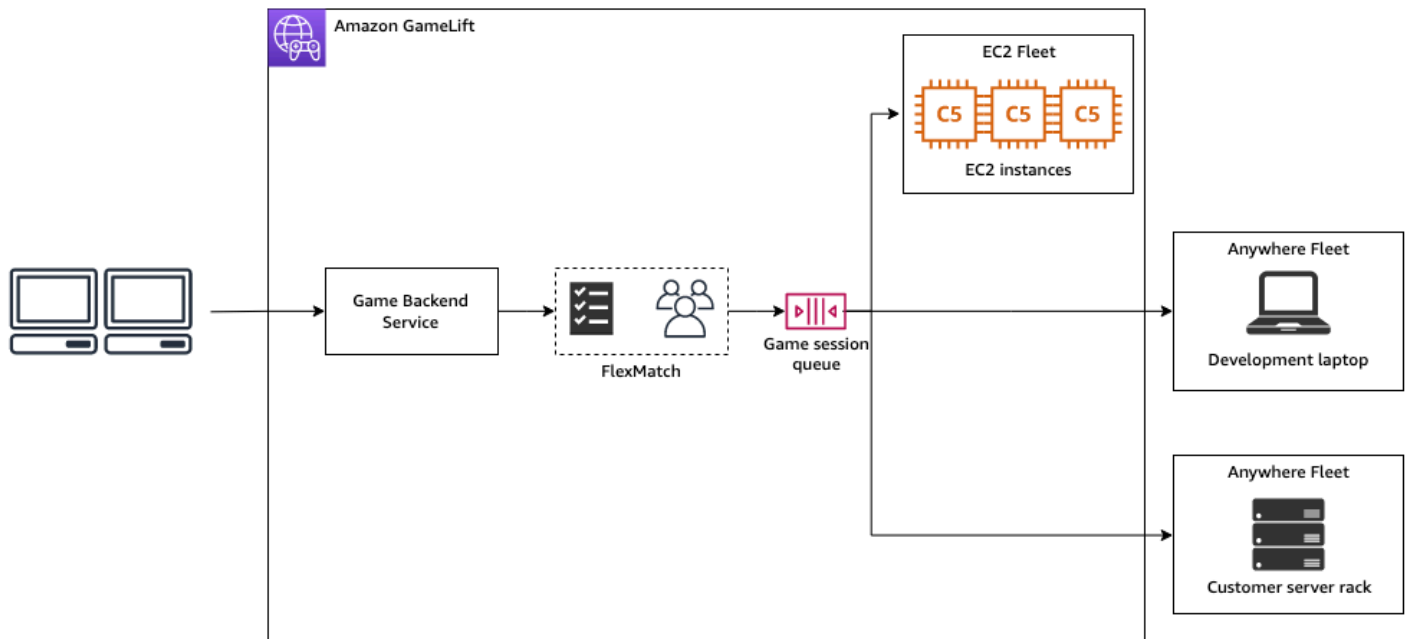
Tokens de autenticação

Quando você inicializa um servidor de jogos em seu computador, o Amazon GameLift Server SDK usa um token de autenticação para autenticar seu servidor de jogos na Amazon. GameLift É possível reutilizar o mesmo token de autenticação para todos os servidores de jogos no mesmo computador, até o prazo de expiração do token de autenticação. Para recuperar o token de autenticação, chame o comando [get-compute-auth-token](#) AWS Command Line Interface (AWS CLI). Passe o token para cada servidor do jogo conforme necessário.

Sessões de jogos

Cada sessão de jogo em um computador usa o mesmo token de autenticação criado ao registrar o computador em um local da frota.

O diagrama a seguir mostra uma fila de sessões de jogo que usa FlexMatch matchmaking e várias frotas. As frotas incluem uma frota EC2 com instâncias C5, uma frota Anywhere com um laptop de desenvolvimento e uma frota Anywhere com um rack de servidor hospedado pelo cliente.



Tópicos

- [Criar uma política personalizada](#)
- [Criar uma frota](#)
- [Registrar seu computador](#)
- [Executar um processo de servidor](#)
- [Criar sessões de jogos](#)
- [Migre para o EC2 gerenciado](#)

Criar uma política personalizada

Para começar a hospedar jogos em seus recursos de computação, crie um local personalizado descrevendo onde o computador reside.

Console

Para criar uma política personalizada

1. Abra o [GameLift console da Amazon](#).
2. No painel de navegação, em Hospedagem, escolha Locais.
3. Na página Locais, selecione Criar local.
4. Na caixa de diálogo Criar local, faça o seguinte:

- a. Insira um Nome de localização. Isso rotula a localização do seu hardware que a Amazon GameLift usa para executar seus jogos em Anywhere frotas. A Amazon GameLift acrescenta o nome do seu local personalizado com custom-.
- b. (Opcional) Adicione tags como pares de chave-valor à sua localização personalizada. Escolha Adicionar nova tag para cada tag a ser adicionada.
- c. Escolha Criar.

AWS CLI

Crie um local personalizado usando o comando [create-location](#). Os location-name rótulos indicam a localização do seu hardware que a Amazon GameLift usa para executar seus jogos em Anywhere frotas. Ao criar o local personalizado, o nome do local deve começar com custom-.

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

Saída

```
{  
  "Location": {  
    "LocationName": "custom-location-1",  
    "LocationArn": "arn:aws:gamelift:us-east-1:111122223333:location/custom-  
location-1"  
  }  
}
```

Criar uma frota

Use o [GameLift console da Amazon](#) ou o AWS CLI para criar uma Anywhere frota.

Depois de criar uma nova frota Anywhere, o status da frota passa de NEW para ACTIVE. Quando atinge o status é ACTIVE, a frota está pronta para hospedar sessões de jogo. Para obter ajuda com problemas de criação de frotas, consulte [Depurar problemas com a frota do Amazon GameLift](#).

Console

Para criar uma frota Anywhere

1. Abra o [GameLift console da Amazon](#).
2. No painel de navegação, em Hospedagem, selecione Frotas.
3. Na página Frotas, selecione Criar frota.
4. Na etapa Tipo de computação, selecione Anywhere e, em seguida, selecione Avançar.
5. Na etapa Detalhes da frota, defina os detalhes e selecione Avançar.
6. Na etapa Locais personalizados, selecione o local personalizado criado e selecione Avançar. A Amazon seleciona GameLift automaticamente a casa Região da AWS como a região em que você está criando a frota. Use a região de origem para acessar e usar seus recursos.
7. Conclua as etapas restantes da criação da frota e selecione Enviar para criar sua frota Anywhere.

AWS CLI

Crie uma frota Anywhere usando o comando [create-fleet](#). Inclua sua localização personalizada em `locations`. GameLift A Amazon cria a frota na sua região de origem e nos locais personalizados que você fornece. No exemplo a seguir, substitua *FleetName* e *custom-location-1* por suas próprias informações. A variável `custom-location-1` é o nome do local criado na etapa [Criar uma política personalizada](#).

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

Exemplo de saída

```
{  
  "FleetAttributes": {  
    "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",  
    "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445",  
    "Name": "HardwareAnywhere",  
    "CreationTime": "2023-02-23T17:57:42.293000+00:00",  
    "Status": "ACTIVE",
```

```

    "MetricGroups": [
      "default"
    ],
    "CertificateConfiguration": {
      "CertificateType": "DISABLED"
    },
    "ComputeType": "ANYWHERE"
  }
}

```

Registrar seu computador

Para registrar seu recurso de computação na frota criada, use o comando [register-compute](#). Substitua o *fleet-id* pelo fleet-id devolvido na etapa anterior ou pelo ARN da frota encontrado na página de detalhes da sua frota no console. Substitua o *compute-name* e *ip-address* pelo endereço IP do seu recurso de computação.

Note

Recomendamos chamar os get-compute-auth-token comandos register-compute e a partir de um script ou gerenciador de processos separado do servidor do jogo.

```

aws gamelift register-compute \
  --compute-name HardwareAnywhere \
  --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445 \
  --ip-address 10.1.2.3 \
  --location custom-location-1

```

Exemplo de saída

```

{
  "Compute": {
    "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",
    "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445",
    "ComputeName": "HardwareAnywhere",
    "ComputeArn": "arn:aws:gamelift:us-east-1:111122223333:compute/  
HardwareAnywhere",
  }
}

```

```
"IpAddress": "10.1.2.3",
"ComputeStatus": "Active",
"Location": "custom-location-1",
"CreationTime": "2023-02-23T18:09:26.727000+00:00",
"GameLiftServiceSdkEndpoint": "wss://us-east-1.api.amazongamelift.com"
}
}
```

Executar um processo de servidor

1. Obtenha o token de autenticação para o recurso de computação da frota criada.

Seu servidor de jogo usa o token de autenticação para se autenticar na Amazon GameLift. Cada token de autenticação tem um prazo de validade. Para continuar usando o recurso de computação para hospedar o servidor de jogos, recupere um novo token de autenticação antes da expiração.

Note

GameLift A Amazon recomenda chamar os `get-compute-auth-token` comandos `register-compute` e a partir de um script ou gerenciador de processos separado do seu servidor de jogo.

No exemplo a seguir, substitua `fleet-id` o pelo ARN ou ID da frota criada nas etapas anteriores. Substitua o `compute-name` pelo nome da computação criada usando o comando `register-compute` em uma etapa anterior.

```
aws gamelift get-compute-auth-token \
  --fleet-id arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-
cebb4da2-52a8-4c27-9b85-587f945c6445 \
  --compute-name HardwareAnywhere
```

Resultado do exemplo:

```
{
  "FleetId": "fleet-cebb4da2-52a8-4c27-9b85-587f945c6445",
  "FleetArn": "arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-
cebb4da2-52a8-4c27-9b85-587f945c6445",
  "ComputeName": "HardwareAnywhere",
```

```

    "ComputeArn": "arn:aws:gamelift:us-east-1:111122223333:compute/
HardwareAnywhere",
    "AuthToken": "0c728041-3e84-4aaa-b927-a0fb202684c0",
    "ExpirationTimestamp": "2023-02-23T18:47:54+00:00"
}

```

2. Execute uma instância do executável do servidor de jogos.

Para executar o servidor de jogos, inicialize seu servidor de jogos chamando `InitSDK()` e transmitindo seus parâmetros de servidor. Para ter mais informações, consulte [Parâmetros do servidor](#).

Entrada do SDK do servidor:

```

//Define the server parameters
ServerParameters serverParameters = new ServerParameters(
    websocketUrl=wss://us-east-1.api.amazongamelift.com,
    processId=PID1234,
    hostId=HardwareAnywhere,
    fleetId=arn:aws:gamelift:us-east-1:111122223333:fleet/fleet-
cebb4da2-52a8-4c27-9b85-587f945c6445,
    authToken=0c728041-3e84-4aaa-b927-a0fb202684c0);

//InitSDK establishes a connection with GameLift's websocket server for
communication.
var initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);

```

3. Depois que o processo do servidor estiver pronto para hospedar uma sessão de jogo, ligue `ProcessReady()` do seu servidor de jogos para a Amazon GameLift. Para obter mais informações sobre parâmetros de processo, consulte [ProcessParameters](#)

```

// Set parameters and call ProcessReady
var processParams = new ProcessParameters(
    this.OnStartGameSession,
    this.OnProcessTerminate,
    this.OnHealthCheck,
    this.OnUpdateGameSession,
    port=1024,
    new LogParameters(new List<string>()           // Examples of log and error files
written by the game server
    {
        "C:\\game\\logs",
        "C:\\game\\error"
    }

```

```
    })  
  );  
  
  var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);
```

Criar sessões de jogos

1. Adicione lógica ao servidor de jogo para que o processo do servidor responda à mensagem `onStartGameSession()` com `ActivateGameSession()`. Essa operação não tem parâmetros, mas envia uma confirmação à Amazon de GameLift que seu servidor recebeu e aceitou a mensagem de criação da sessão de jogo.

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

2. No serviço de back-end do seu cliente de jogo, inicie sua sessão de jogo usando o comando [start-matchmaking](#), [start-game-session-placement](#) ou [create-game-session](#).

```
aws gamelift create-game-session \  
  --fleet-id arn:aws:gamelift:us-east-1:682428703967:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445 \  
  --name GameSession1 \  
  --maximum-player-session-count 2 \  
  --location custom-location-1
```

Resultado do exemplo:

```
GameSession {  
  FleetId = arn:aws:gamelift:us-east-1:682428703967:fleet/fleet-  
cebb4da2-52a8-4c27-9b85-587f945c6445,  
  GameSessionId = 4444-4444,  
  Name = GameSession1,  
  Location = custom-location-1,  
  IPAddress = 10.2.3.4,  
  Port = 1024,  
  ...
```

```
}
```

GameLift A Amazon envia uma `onStartGameSession()` mensagem para seu processo de servidor registrado. A mensagem contém o objeto `GameSession` da etapa anterior com propriedades do jogo, dados das sessões do jogo, dados do marcador de jogos e mais sobre a sessão do jogo.

3. Quando a sessão do jogo estiver concluída, finalize o processo do servidor do jogo.

Entrada do SDK do servidor:

```
var processEndingOutcome = GameLiftServerAPI.ProcessEnding();
if (processReadyOutcome.Success)
    Environment.Exit(0);
// otherwise, exit with error code
Environment.Exit(errorCode);
```

4. Inicie outro processo do servidor de jogos ao chamar `ProcessReady(processParams)`.

Migre para o EC2 gerenciado

Depois de desenvolver seu servidor de jogos e estar pronto para se preparar para a produção, você pode fazer com que a Amazon GameLift gerencie seu hardware. Para migrar para uma frota EC2 gerenciada, faça o upload da sua compilação para a Amazon GameLift e crie uma frota EC2 gerenciada. Para obter mais informações sobre como fazer o upload de sua compilação e configurar uma frota, consulte [Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift](#) e [Criar uma frota gerenciada pelo Amazon GameLift](#).

Gerencie suas frotas do Amazon GameLift

Use o console do Amazon GameLift ou a CLI da AWS para atualizar as configurações da sua frota, alterar locais remotos ou excluir uma frota.

Atualizar uma configuração de frota

Atualize atributos mutáveis da frota, configurações de porta e configurações de runtime usando o console do Amazon GameLift ou a CLI da AWS. Para alterar os limites de escala, consulte [Capacidade de frota em ajuste de escala automático com o Amazon GameLift](#).

Amazon GameLift console

1. No [console do Amazon GameLift](#), no painel de navegação, selecione Frotas.
2. Selecione a frota que deseja atualizar. Uma frota deve ter o status ACTIVE antes de poder editá-la.
3. Na página de detalhes da frota, em qualquer uma das seções a seguir, escolha Editar.
 - Configurações da frota
 - Altere os atributos da frota, como Nome e Descrição.
 - Adicione ou remova Grupos de métricas, que o Amazon CloudWatch usa para rastrear métricas agregadas do Amazon GameLift para várias frotas.
 - Atualize as configurações de Limite de criação de recursos.
 - Ative ou desative a proteção da sessão de jogo.
 - Configuração de runtime – Alterar qualquer uma das seguintes configurações de runtime e adicionar ou remover configurações de runtime.
 - Altere o Caminho de inicialização do seu servidor de jogo.
 - Adicione, remova ou altere os Parâmetros opcionais de inicialização.
 - Altere o número de Processos simultâneos que seus servidores de jogos executam.
 - Ativação da sessão de jogo – Altere a forma como você deseja que os processos do servidor sejam executados e hospedem sessões de jogo atualizando o Máximo de ativações simultâneas de sessões de jogo e o Novo tempo limite de ativação.
 - Configurações de porta EC2 – Atualize o endereço IP e os intervalos de portas que permitem acesso de entrada à frota.
4. Depois, selecione Confirmar para salvar as alterações.

AWS CLI

Use os seguintes comandos da CLI da AWS para atualizar uma frota:

- [update-fleet-attributes](#)
- [update-fleet-port-settings](#)
- [update-runtime-configuration](#)

Atualizar localizações da frota

É possível adicionar ou remover os locais remotos de uma frota usando o console do Amazon GameLift ou a CLI da AWS. Não é possível alterar a região de origem de uma frota.

Amazon GameLift console

1. No [console do Amazon GameLift](#), no painel de navegação, selecione Frotas.
2. Selecione a frota que deseja atualizar. Uma frota deve ter o status ACTIVE antes de poder editá-la.
3. Na página de detalhes da frota, escolha o guia Localizações para visualizar as localizações da frota.
4. Para adicionar novos locais remotos, escolha Adicionar e selecione os locais nos quais você deseja implantar instâncias. Essa lista não inclui instâncias em que o tipo de instância da frota não está disponível.
5. Com os novos locais selecionados, escolha Adicionar. O Amazon GameLift adiciona os novos locais à lista, com o status definido como NEW. Em seguida, o Amazon GameLift começa a provisionar uma instância em cada local adicionado e a prepará-la para hospedar sessões de jogo.
6. Para remover locais remotos existentes da frota, use as caixas de seleção para selecionar um ou mais locais listados.
7. Com uma ou mais frotas selecionadas, escolha Remover. Os locais removidos permanecem na lista, com o status definido como DELETING. Em seguida, o Amazon GameLift inicia o processo de encerramento da atividade no local removido. Se houver instâncias ativas que estejam hospedando sessões de jogo, o Amazon GameLift usa o processo de encerramento do servidor de jogos para finalizar as sessões de jogo, encerrar servidores de jogos e desligar instâncias sem problemas.

AWS CLI

Use os seguintes comandos da CLI da AWS para atualizar locais de frota:

- [create-fleet-locations](#)
- [delete-fleet-locations](#)

Excluir uma frota

É possível excluir uma frota quando não precisar mais dela. A exclusão de uma frota remove permanentemente todos os dados associados às sessões de jogos e de jogadores e dados de métricas coletados. Como alternativa, é possível reter a frota, desabilitar o ajuste de escala automático e escalar a frota manualmente para 0 instância.

Note

Caso a frota tenha uma conexão de emparelhamento da VPC, primeiro é necessário solicitar autorização chamando [CreateVpcPeeringAuthorization](#). O Amazon GameLift exclui a conexão de emparelhamento da VPC durante a exclusão da frota.

É possível usar o console do Amazon GameLift ou a ferramenta da CLI da AWS para excluir uma frota.

Amazon GameLift console

1. No [console do Amazon GameLift](#), no painel de navegação, selecione Frotas.
2. Escolha a frota que você deseja excluir. Você só pode excluir frotas no status ACTIVE ou ERROR.
3. Escolha Excluir.
4. Na caixa de diálogo Excluir frota, confirme a exclusão inserindo **delete**.
5. Selecione a opção Excluir.

AWS CLI

Use os seguintes comandos da CLI da AWS para excluir um frota:

- [delete-fleet](#)

Adicionar um alias a uma frota da Amazon GameLift

Um GameLift alias da Amazon é usado para abstrair a designação de uma frota. As designações de frota informam à Amazon GameLift onde pesquisar os recursos disponíveis ao criar novas sessões de jogo para os jogadores. Use aliases em vez de IDs de frota específica para alternar o tráfego do

jogador de uma frota para outra de forma mais fácil e transparente, alterando o local de destino do alias.

Há dois tipos de estratégias de roteamento para aliases:

- Simples – Roteia o tráfego de jogadores para o ID de uma frota especificada. Você pode atualizar o ID da frota para um alias a qualquer momento.
- Terminal – Reenvia uma mensagem para o cliente. Por exemplo, você pode direcionar jogadores que estão usando um out-of-date cliente para um local onde eles possam obter um upgrade.

As frotas têm uma vida útil finita e há várias razões para trocá-las durante a vida de um jogo. Não é possível atualizar a compilação do servidor de jogos da frota ou alterar certos atributos de recursos de computação em uma frota existente. Em vez disso, crie novas frotas com as alterações e, em seguida, alternar os jogadores para as novas frotas. Com aliases, a mudança de frotas tem impacto mínimo sobre o seu jogo e é invisível para os jogadores.

Os aliases são úteis em jogos que não usam filas. A troca de frotas em uma fila é uma simples questão de criar uma nova frota, adicioná-la à fila e remover a antiga, sem que nada disso fique visível para os jogadores. Por outro lado, os clientes de jogos que não usam filas devem especificar qual frota usar ao se comunicar com o serviço da Amazon GameLift . Sem aliases, uma troca de frota exige atualizações no código do jogo e, possivelmente, distribuição de clientes de jogos atualizados aos jogadores.

Ao atualizar o ID da frota para o qual um alias aponta, há um período de transição de até 2 minutos em que as sessões de jogo com o alias podem acabar na frota antiga.

Criar um novo alias

[Você pode criar um alias usando o GameLift console da Amazon, conforme descrito aqui, ou com o comando da AWS CLI create-alias.](#)

1. No [GameLift console da Amazon](#), no painel de navegação, escolha Aliases.
2. Na página Aliases, escolha Criar alias. É recomendável incluir o tipo de frota em seus nomes de alias. Isso facilita muito a identificação do tipo de frota ao visualizar uma lista de aliases.
3. Na página Criar alias, em Detalhes do alias, faça o seguinte:
 - a. Em Nome, insira um nome de alias.
 - b. Para descrição, insira uma breve descrição para identificação.

- c. Escolha o tipo de roteamento simples ou terminal.
4. (Opcional) Em Tags, adicione tags ao alias inserindo pares de Chave e Valor.
5. Escolha Criar.

Editar um alias

[Você pode editar um alias usando o GameLift console da Amazon ou com o comando da AWS CLI `update-alias`.](#)

1. No [GameLift console da Amazon](#), no painel de navegação, escolha Aliases.
2. Na página Aliases, selecione o alias a ser editado.
3. Na página de alias, escolha Editar.
4. Na página Editar alias, você pode editar o seguinte:
 - Nome do alias – Nome amigável para o alias.
 - Descrição – Descrição resumida do alias.
 - Tipo – Estratégia de roteamento para tráfego de jogadores. Selecione Simples para alterar a frota associada ou selecione Terminal para editar a mensagem de encerramento.
5. Escolha Salvar alterações.

Depurar problemas com a frota do Amazon GameLift

Este tópico fornece orientações sobre problemas de configuração de frota para uma solução gerenciada do Amazon GameLift. Para obter uma solução de problemas adicional, é possível acessar remotamente uma instância de frota quando a frota estiver ativa. Consulte [Conecte-se remotamente às instâncias da GameLift frota da Amazon](#).

Depurar problemas de criação de frotas

Quando uma frota é criada, o serviço Amazon GameLift inicia um fluxo de trabalho que implanta uma nova instância em cada local da frota e a prepara para executar os servidores de jogos. Para obter uma descrição detalhada, consulte [Como funciona a criação de frota do Amazon GameLift](#). Uma frota não pode hospedar sessões de jogo e jogadores até atingir o status Ativo. Esta seção discute os problemas mais comuns que impedem que as frotas se tornem ativas.

Download e validação

Durante essa fase, a criação da frota pode falhar se houver problemas com os arquivos de compilação extraídos, se não for possível executar o script de instalação, ou se os executáveis designados na configuração do runtime não forem incluídos nos arquivos de compilação. O Amazon GameLift fornece logs relacionados a cada um desses problemas.

Se os logs não revelaram um problema, é possível que o problema tenha ocorrido devido a um erro interno de serviço. Nesse caso, tente criar a frota novamente. Se o problema persistir, faça upload da compilação de jogo novamente (caso os arquivos estejam corrompidos). Também é possível entrar em contato com o suporte do Amazon GameLift ou publicar uma pergunta no fórum.

Desenvolvimento

Problemas que causam falha durante a fase de compilação muitas vezes ocorrem devido a problemas com os arquivos de compilação de jogo e/ou o script de instalação. Observe que os arquivos de compilação do jogo, após o upload para o Amazon GameLift, podem ser instalados em um computador com o sistema operacional adequado. Use uma instalação limpa do sistema operacional, não um ambiente de desenvolvimento existente.

Ativação

Os problemas mais comuns na criação de frotas ocorrem durante a fase de Ativação. Durante essa fase uma série de elementos estão sendo testados, incluindo a viabilidade do servidor de jogos, as definições de configuração em runtime e a capacidade do servidor de jogos usar o SDK de servidor para interagir com o serviço do Amazon GameLift. Problemas comuns que surgem durante a ativação de frotas incluem:

Há falha na inicialização de processos do servidor.

Primeiro, verifique se você definiu corretamente o caminho de inicialização e os parâmetros de inicialização opcionais na configuração do tempo de execução da frota. Visualize a configuração de runtime atual da frota usando a página de detalhes da frota, seção [Detalhes](#) ou chamando o comando [describe-runtime-configuration](#) da AWS CLI. Se a configuração do tempo de execução estiver correta, verifique se há problemas com os arquivos de compilação do jogo e/ou o script de instalação.

Os processos do servidor começam, mas a frota deixa de ser ativada.

Se os processos do servidor forem iniciados e executados com êxito, mas o status da frota não for alterado para Ativa, é provável que o processo do servidor não esteja notificando o Amazon GameLift de que ele está pronto para hospedar as sessões de jogos. Verifique se o servidor de

jogos está chamando corretamente a ação de `Server API ProcessReady()` (consulte [Inicializar o processo do servidor](#)).

Falha na solicitação de conexão de emparelhamento de VPCs.

Para frotas que são criadas com uma conexão de emparelhamento de VPCs (consulte [Como configurar o emparelhamento de VPC com uma nova frota](#)), o emparelhamento de VPCs é feito durante essas fases de Ativação. Se um emparelhamento de VPCs falhar por qualquer motivo, haverá falha na mudança da nova frota para o status Ativo. Você pode rastrear o sucesso ou a falha da solicitação de emparelhamento chamando [describe-vpc-peering-connections](#). Certifique-se de verificar se existe uma autorização de emparelhamento de VPCs válida ([describe-vpc-peering-authorizations](#)), uma vez que as autorizações são válidas por apenas 24 horas.

Problemas de processo do servidor

Os processos do servidor começam, mas falham rapidamente ou apresentam integridade insatisfatória.

Outros problemas com a compilação do jogo, podendo acontecer quando se tenta executar muitos processos de servidor simultaneamente na instância. O número ideal de processos simultâneos depende do tipo de instância e dos requisitos de recursos do servidor de jogos. Tente reduzir o número de processos simultâneos, algo definido na configuração de tempo de execução da frota, para ver se o desempenho melhora. É possível alterar a configuração de runtime de uma frota usando o console do Amazon GameLift (editar as configurações de alocação da capacidade da frota) ou chamar o comando [update-runtime-configuration](#) da AWS CLI.

Problemas de exclusão de frota

Uma frota não pode ser encerrada devido à contagem máxima de instâncias.

A mensagem de erro indica que a frota que está sendo excluída ainda tem instâncias ativas, o que não é permitido. Você deve primeiro reduzir uma frota para zero instâncias ativas. Isso é feito definindo manualmente a contagem de instâncias desejada da frota para "0" e aguardando até que a redução entre em vigor. Desligue a escalabilidade automática, pois ela neutralizará as configurações manuais.

As ações da VPC não estão autorizadas.

Esse problema se aplica somente às frotas para as quais você criou especificamente conexões de emparelhamento de VPC (consulte [Emparelhamento de VPC para o Amazon GameLift](#)). Esse cenário ocorre porque o processo de exclusão de uma frota também inclui a exclusão da VPC da frota e de qualquer conexão de emparelhamento de VPC. Primeiro, obtenha uma autorização chamando a API [CreateVpcPeeringAuthorization\(\)](#) do serviço do Amazon GameLift ou usando o comando `create-vpc-peering-authorization` da CLI da AWS. Quando tiver a autorização, você pode excluir a frota.

Problemas com a frota de servidores em tempo real

Sessões de jogos zumbi: um jogo é iniciado e executado, mas nunca termina.

Você pode observar esses problemas como qualquer um dos seguintes cenários:

- Atualizações de script não são selecionadas por servidores em tempo real da frota.
- A frota atinge a capacidade máxima rapidamente e não reduz quando as atividades dos jogadores (como novas solicitações de sessão de jogo) é reduzida.

Isso é quase certamente um resultado de não chamar `processEnding` com êxito no script em tempo real. Embora a frota se torne ativa e as sessões de jogos sejam iniciadas, não há um método para interrompê-las. Como resultado, o servidor em tempo real que está executando a sessão de jogo nunca é liberado para iniciar uma nova, e as novas sessões de jogos só podem ser iniciadas quando novos servidores em tempo real forem rotacionados. Além disso, as atualizações no script em tempo real não afetam as sessões de jogo que já estão em execução, apenas as novas.

Para impedir que isso aconteça, os scripts precisam fornecer um mecanismo para acionar uma chamada de `processEnding`. Conforme ilustrado no [Exemplo de script de servidores em tempo real](#), uma maneira é programar um tempo limite de sessão ociosa em que, se nenhum jogador estiver conectado por um determinado período de tempo, o script encerrará a sessão do jogo atual.

No entanto, se você se encaixar nesse cenário, há duas alternativas para destravar os servidores em tempo real. O truque é acionar os processos do servidor em tempo real (ou as instâncias da frota subjacentes) para reiniciar. Nesse caso, o GameLift fecha automaticamente as sessões de jogos para você. Assim que os servidores em tempo real são liberados, eles podem iniciar novas sessões de jogos usando a versão mais recente do script em tempo real.

Há dois métodos para fazer isso, dependendo de quão disseminado é o problema:

- Reduza toda a frota. Esse método é o mais simples de executar, mas tem um efeito amplo. Reduza a frota para zero instância, aguarde até que a frota seja totalmente reduzida e expanda-a novamente. Isso limpará todas as sessões de jogos existentes e permitirá que você comece de novo com o script em tempo real atualizado mais recentemente.
- Acesse a instância remotamente e reinicie o processo. Esta será uma boa opção se você tiver apenas alguns processos para corrigir. Se você já estiver conectado à instância, como para logs finais ou depuração, esse poderá ser o método mais rápido. Consulte [Conecte-se remotamente às instâncias da GameLift frota da Amazon](#).

Caso opte por não incluir uma maneira de chamar `processEnding` no script em tempo real, haverá algumas situações delicadas que poderão ocorrer, mesmo quando a frota estiver ativa e sessões de jogos estiverem iniciadas. Primeiro, uma sessão de jogo em execução não termina. Como resultado, o processo do servidor que está executando a sessão do jogo nunca é liberado para iniciar uma nova sessão de jogo. Em segundo lugar, o servidor em tempo real não recebe nenhuma atualização de script.

Conecte-se remotamente às instâncias da GameLift frota da Amazon

Você pode se conectar a qualquer instância em suas frotas ativas do EC2 GameLift gerenciadas pela Amazon. Os motivos comuns para acessar uma instância incluem:

- Solucione problemas com a integração do seu servidor de jogos
- Ajuste sua configuração de tempo de execução e outras configurações específicas da frota
- Obtenha atividades do servidor de jogos em tempo real, como rastreamento de registros.
- Execute ferramentas de benchmarking usando o tráfego real de jogadores.
- Investigue problemas específicos com uma sessão de jogo ou processo do servidor.

Ao se conectar a uma instância, considere estes possíveis problemas:

- Você pode se conectar a instâncias em frotas ativas. As frotas não ativas, aquelas que estão sendo ativadas ou estão em estado de erro, podem ficar acessíveis por um curto período de tempo. Para obter ajuda com problemas de ativação da frota, consulte [Depurar problemas com a frota do Amazon GameLift](#).
- A conexão com uma instância ativa não afeta a atividade de hospedagem da instância. A instância continua iniciando e interrompendo os processos do servidor com base na configuração do tempo

de execução. Ele ativa e hospeda a sessão do jogo. Ele pode ser encerrado em resposta a um evento de redução ou outro evento.

- Qualquer alteração feita nos arquivos ou nas configurações da instância pode afetar as sessões de jogo ativas da instância e os jogadores conectados.

As instruções a seguir descrevem como se conectar remotamente a uma instância usando a interface de linha de AWS comando (CLI). Você também pode fazer chamadas programáticas usando o AWS SDK, conforme documentado na referência da [API de GameLift serviços da Amazon](#).

Colete dados da instância

Colete as seguintes informações:

- O ID da instância à qual você deseja se conectar. Você pode usar o ID da instância ou o ARN.
- A versão GameLift do SDK do servidor Amazon que está sendo usada na instância. O SDK do servidor é integrado à compilação do jogo em execução na instância.

Para recuperar dados da instância

As etapas a seguir pressupõem que você tenha um ID de frota EC2 gerenciado para a instância à qual deseja se conectar.

1. Obtenha o nome do computador.

Call [list-compute](#) para a frota gerenciada do EC2 para obter uma lista de todas as computações ativas na frota. Para uma frota de localização única, especifique o ID da frota ou o ARN. Para uma frota com vários locais, especifique o ID da frota ou o ARN e um local. Para uma frota EC2 gerenciada, as computações são instâncias EC2 e a propriedade retornada `ComputeName` é o ID da instância. Por exemplo: .

Solicitação

```
aws gamelift list-compute \  
  --fleet-id "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa" \  
  --location ""sa-east-1"
```

Resposta

```
{
  "ComputeList": [
    {
      "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
      "FleetArn": "arn:aws:gamelift:us-west-2::fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
      "ComputeName": "i-0abc12d3e45fa6b78",
      "IpAddress": "00.00.000.00",
      "DnsName":
      "b08444ki909kvqu6zpw3is24x5pyz4b6m05i3jbxvpk9craztu0lqrbbrbnbkks.uwp57060n1k6dnlnw49b78hg1
west-2.amazongamelift.com",
      "ComputeStatus": "Active",
      "Location": "sa-east-1",
      "CreationTime": "2023-07-09T22:51:45.931000-07:00",
      "OperatingSystem": "AMAZON_LINUX",
      "Type": "c4.large"
    }
  ]
}
```

2. Encontre a versão do SDK do servidor.

A versão do SDK do servidor é um atributo de um recurso de compilação.

- Ligue [describe-fleet-attributes](#) com um ID de frota para obter o ID de construção e o ARN da frota.
- Chame [describe-build](#) com o ID de compilação ou ARN para obter a versão SDK do servidor da compilação.

Por exemplo: .

Solicitação

```
aws gamelift describe-fleet-attributes /
--fleet-ids "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
```

Resposta

```
{
  "FleetAttributes": [
```

```
{
  "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
  "ComputeType": "EC2",
  "BuildId": "build-3333cccc-44dd-55ee-66ff-00001111aa22",
  . . .
}
]
```

Solicitação

```
aws gamelift describe-build /
--build-id "build-3333cccc-44dd-55ee-66ff-00001111aa22"
```

Resposta

```
"Build": {
  "BuildId": "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
  "Name": "My_Game_Server_Build_One",
  "OperatingSystem": "AMAZON_LINUX_2",
  "ServerSdkVersion": "5.1.1",
  . . .
}
```

Conecte-se a uma instância (SDK 5 do servidor)

Se a instância à qual você deseja se conectar estiver executando uma compilação de jogo com o SDK do servidor versão 5.x, use as instruções a seguir para se conectar à instância usando o Amazon EC2 Systems Manager (SSM). Você pode acessar instâncias remotas que estejam executando o Windows ou o Linux.

1. Solicite credenciais de acesso para a instância. Quando você tiver um nome de computação e um ID de frota para a instância à qual deseja se conectar, ligue [get-compute-access](#). Se for bem-sucedido, a Amazon GameLift retornará um conjunto de credenciais temporárias para acessar a instância. Por exemplo: .

Solicitação

```
aws gamelift get-compute-access \
--compute-name i-11111111a222b333c \
```

```
--fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa
--region us-west-2
```

Resposta

```
{
  "ComputeName": " i-11111111a222b333c ",
  "Credentials": {
    "AccessKeyId": " ASIAIOSFODNN7EXAMPLE ",
    "SecretAccessKey": " wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY ",
    "SessionToken": " AQoDYXdzEJr...<remainder of session token>"
  },
  "FleetArn": " arn:aws:gamelift:us-west-2::fleet/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa ",
  "FleetId": " fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa "
}
```

2. Exporte as credenciais de acesso. Opcionalmente, você pode exportar as credenciais para variáveis de ambiente e usá-las para configurar a AWS CLI para o usuário padrão. Para obter mais detalhes, consulte [Variáveis de ambiente para configurar a AWS CLI](#) no Guia do AWS Command Line Interface usuário.

```
export AWS_ACCESS_KEY_ID=ASIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of session token>
```

3. Conecte-se à instância da frota. Inicie uma sessão de SSM com a instância à qual você quer se conectar. Inclua a AWS região ou o local da instância. Para obter mais informações, consulte [Iniciando uma sessão \(AWSCLI\)](#) no Guia do usuário do Amazon EC2 Systems Manager. Use as credenciais que você adquiriu na Etapa 1. Por exemplo: .

```
aws ssm start-session \
--target i-11111111a222b333c \
--region us-west-2
```

Conecte-se a uma instância (SDK do servidor 4.x ou anterior)

Se a instância à qual você deseja se conectar estiver executando uma compilação de jogo com o SDK do servidor versão 4 ou anterior, use as instruções a seguir. Você pode se conectar a instâncias que estão executando Windows ou Linux. Conecte-se a uma instância do Windows usando um

cliente de protocolo de desktop remoto (RDP). Conecte-se a uma instância Linux usando um cliente SSH.

1. Solicite credenciais de acesso para a instância. Quando você tiver um ID de instância, use o comando [get-instance-access](#) para solicitar credenciais de acesso. Se for bem-sucedida, a Amazon GameLift retornará o sistema operacional, o endereço IP e um conjunto de credenciais da instância (nome de usuário e chave secreta). O formato de credenciais depende do sistema operacional da instância. Use as instruções a seguir para recuperar credenciais de RDP ou SSH.
 - Para instâncias do Windows – Para se conectar a uma instância do Windows, o RDP requer um nome de usuário e uma senha. A solicitação `get-instance-access` retorna esses valores como strings simples, de maneira que você possa usar os valores retornados como estão. Exemplo de credenciais:

```
"Credentials": {
  "Secret": "aA1bBB2cCCd3EEE",
  "UserName": "gl-user-remote"
}
```

- Para instâncias do Linux – Para se conectar a uma instância do Linux, o SSH requer um nome de usuário e uma chave privada. A Amazon GameLift emite chaves privadas RSA e as retorna como uma única string, com o caractere de nova linha (`\n`) indicando quebras de linha. Para tornar a chave privada utilizável, siga estas etapas: (1) converter a string em um `.pem` arquivo e (2) definir permissões para o novo arquivo. Exemplo de credenciais retornadas:

```
"Credentials": {
  "Secret": "-----BEGIN RSA PRIVATE KEY-----
nEXAMPLEKEYKCAQEAY7WZhaDsrA1W3mR1QtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
\nvBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7i05dSrvC7dQkW2duV5QuUdE0QW
\nZ/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449i1x9X1F
\nG50TCFe0zfl8dqqCP6GzbPaIjiU19xX/az0R9V+tpU0zEL+wmXnZt3/nHPQ5xvD20JH67km6SuPW
\noPzev/D8V+x4+bHthfSjR9Y7DvQFjfbVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrrqu
\n/uler7vgIn5m71N5LKw4hJLAIW6tUT/fzvtCHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/
ufGxbL1\nmb5qwmGUNepJaZD6QSSs3kICLwUUYUiGfc0uisbmJoap/
GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2\nbahyWyJNfjLe4M86yd2YK3V2CmK+X/
B0sShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/
tJWSD9\n81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA30zdXzMqexXVJ1TLZVEH0E7bh1Y9d801ozR
\noQs/FiZNAx2iijCWyv01pjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfq1
+1Ip1\nYkriL0DbLXlvRAH+yHPRit2hH0jtUNZh4Axv+cpg09qbUI3+43eEy24B7G/Uh
+GTfbjsXs0xQx/x\np9otyVwc7hsQ5TA5PZb
```

```
+mVkJ50BEKzet9XcKw0NBYELGhnEPe7cCgYEA06Vgov6YH1eHui9kHuws
\nayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWV0Eihvm+xTtmaZlSp//lKq75XDwnU
\nWA8gkn603QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
\nGyBjb0+0Zk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
\noMo7yykABY70zd5wQewBQ4AdS1WSX4nGDtsiFxiI5sKuAAe0CbTosy1s8w8fxoJ5Tz1sdoxNeGs
\nArq6Wv/G16zQuAE9zK9vVwKBgF+09VI/1wJBirsDGz9whVwFFPrTkJNvJZzYt69qezx1sjgFKshy
\nWBhd4xHZtmCqpBP1AymEjr/T01bxyARmXMnIOWIANNXMGB4KGSy11mzSVAoQ+fQr+cJ3d0dyP11j
\nnjjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0i0egLda
\nNWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvT+nY2jVhv7UGd8MjwUTNGItDb6nsYqM2asrnF3qS
\nVRkAKKKYeGjKpUfVTrW0YFjXkfcR/V+QFL50ndHAKJXjW7a4ejJLncTzmZSpYzwApc=\n-----END
RSA PRIVATE KEY-----",
  "UserName": "gl-user-remote"
}
```

Ao usar a AWS CLI, você pode gerar automaticamente um `.pem` arquivo incluindo os parâmetros `--query` e `--output` em sua solicitação. `get-instance-access`

Para definir permissões no arquivo `.pem`, execute o comando a seguir:

```
$ chmod 400 MyPrivateKey.pem
```

- Abra uma porta para a conexão remota. Você pode acessar instâncias nas GameLift frotas da Amazon por meio de qualquer porta autorizada na configuração da frota. Você pode visualizar as configurações de porta da frota usando o comando [describe-fleet-port-settings](#).

Como melhor prática, recomendamos só abrir portas para acesso remoto quando você precisar delas e fechá-las quando terminar. Você não pode atualizar as configurações da porta depois de criar uma frota, mas antes que ela esteja ativa. Se você ficar preso, recrie a frota com as configurações da porta abertas.

Use o comando [update-fleet-port-settings](#) para adicionar uma configuração de porta para a conexão remota (como 22 para SSH ou 3389 para RDP). Para o valor do intervalo de IPs, especifique os endereços IP dos dispositivos que você pretende usar para se conectar (convertidos em formato CIDR). Exemplo:

```
$ AWS gamelift update-fleet-port-settings
--fleet-id "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
--inbound-permission-authorizations
"FromPort=22,ToPort=22,IpRange=54.186.139.221/32,Protocol=TCP"
```

O exemplo a seguir abre a porta 3389 em uma frota do Windows

```
$ AWS gamelift update-fleet-port-settings
--fleet-id "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
  --inbound-permission-authorizations
    "FromPort=3389,ToPort=3389,IpRange=54.186.139.221/32,Protocol=TCP"
```

3. Abra um cliente de conexão remota. Use Área de Trabalho Remota para Windows ou SSH para instâncias do Linux. Conecte-se à instância usando o endereço IP, a configuração de porta e as credenciais de acesso.

Exemplo de SSH:

```
ssh -i MyPrivateKey.pem gl-user-remote@192.0.2.0
```

Visualize arquivos em instâncias remotas

Quando conectado a uma instância remotamente, você tem acesso administrativo e de usuário total. Isso significa que você também pode causar erros e falhas na hospedagem de jogos. Se a instância estiver hospedando jogos com jogadores ativos, você corre o risco de travar as sessões de jogo e perder jogadores, ou interromper os processos de encerramento do jogo e causar erros nos dados e registros salvos do jogo.

Procure esses recursos em uma instância de hospedagem:

- Arquivos de compilação do jogo. Esses arquivos são a versão do jogo que você enviou para a Amazon GameLift. Eles incluem um ou mais executáveis, ativos e dependências do servidor de jogos. Os arquivos de compilação do jogo estão em um diretório raiz chamado `game`:
 - No Windows: `c:\game`
 - No Linux: `/local/game`
- Arquivos de log do jogo. Encontre os arquivos de log que seu servidor de jogo gera no diretório `game` raiz em qualquer caminho de diretório que você designou.
- Recursos de GameLift hospedagem da Amazon. O diretório raiz `Whitewater` contém arquivos usados pelo GameLift serviço da Amazon para gerenciar a atividade de hospedagem de jogos. Não modifique esses arquivos por nenhum motivo.
- Configuração de runtime. Não acesse a configuração de tempo de execução para instâncias individuais. Para fazer alterações em uma propriedade de configuração de tempo de execução,

atualize a configuração de tempo de execução da frota (consulte a operação do AWS SDK [UpdateRuntimeConfiguration](#) ou a AWS CLI [update-runtime-configuration](#)).

- Dados da frota. Um arquivo JSON contém informações sobre a frota à qual a instância pertence, para uso pelos processos do servidor em execução na instância. O arquivo JSON está no seguinte local:
 - No Windows: `C:\GameMetadata\gamelift-metadata.json`
 - No Linux: `/local/gamemetadata/gamelift-metadata.json`
- Certificados TLS. Se a instância estiver em uma frota com a geração de certificados TLS ativada, procure os arquivos de certificado, incluindo o certificado, a cadeia de certificados, a chave privada e o certificado raiz no seguinte local:
 - No Windows: `c:\GameMetadata\Certificates`
 - No Linux: `/local/gamemetadata/certificates/`

Escalar a capacidade de hospedagem do Amazon GameLift

A capacidade de hospedagem, medida em instâncias, representa o número de sessões de jogo que o Amazon GameLift pode hospedar simultaneamente e o número de jogadores simultâneos que essas sessões de jogo podem acomodar. Uma das tarefas mais desafiadoras da hospedagem de jogos é escalar a capacidade para atender à demanda dos jogadores sem desperdiçar dinheiro com recursos que não sejam necessários. Para obter mais informações, consulte [Escalar a capacidade de frota](#).

A capacidade é ajustada no nível de localização da frota. Todas as frotas têm pelo menos um local: a região da AWS de origem da frota. Ao visualizar ou escalar a capacidade, as informações são listadas por local, incluindo a região de origem da frota e quaisquer locais remotos adicionais.

Defina manualmente o número de instâncias a serem mantidas ou configure o ajuste de escala automático para ajustar dinamicamente a capacidade à medida que a demanda dos jogadores muda. É recomendável começar ativando a opção de ajuste de escala automático baseado em destino. O objetivo do ajuste de escala automático com base em origem é manter recursos de hospedagem suficientes para acomodar os jogadores atuais, além de um pouco mais para lidar com picos inesperados na demanda dos jogadores. Para a maioria dos jogos, o ajuste de escala automático com base em origem, oferece uma solução de escalabilidade altamente eficaz.

Os tópicos nesta seção fornecem ajuda detalhada com as seguintes tarefas:

- [Definir limites mínimos e máximos para escalabilidade de capacidade](#)

- [Definir manualmente níveis de capacidade](#)
- [Use o ajuste de escala automático com base em origem](#)
- [Gerenciar o ajuste de escala automático com base em regras \(recurso avançado\)](#)
- [Desativar temporariamente o ajuste de escala automático](#)

Realize a maioria das atividades de escalabilidade de frota usando o console do Amazon GameLift. Também é possível usar um SDK da AWS ou o AWS Command Line Interface (AWS CLI) com a API de serviço do [Amazon GameLift](#).

Para gerenciar a capacidade da frota no console

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, selecione Hospedagem, Frotas.
3. Na página Frotas, escolha o nome de uma frota ativa para abrir a página de detalhes da frota.
4. Escolha a guia Escalabilidade. Neste guia, você poderá:
 - Veja métricas históricas de escalabilidade para toda a frota.
 - Visualize e atualize as configurações de capacidade para cada localização da frota, incluindo limites de escala e configurações de capacidade atuais.
 - Atualize o ajuste de escala automático com base em destino, visualize as políticas de ajuste de escala automático com base em regras aplicadas a toda a frota e suspenda a atividade de ajuste de escala automático em cada local.

Tópicos

- [Defina os limites de capacidade do Amazon GameLift](#)
- [Defina manualmente a capacidade de uma frota do Amazon GameLift](#)
- [Capacidade de frota em ajuste de escala automático com o Amazon GameLift](#)

Defina os limites de capacidade do Amazon GameLift

Ao escalar a capacidade de hospedagem para um local de frota do Amazon GameLift, manualmente ou por ajuste de escala automático, considere os limites de escalabilidade do local. Todas as localizações da frota têm um limite mínimo e máximo que definem o alcance permitido para a

capacidade do local. Por padrão, os limites nos locais da frota têm um mínimo de 0 instâncias e um máximo de 1 instância. Antes de escalar a localização de uma frota, ajuste os limites.

Caso esteja usando o ajuste de escala automático, o limite máximo permite que o Amazon GameLift amplie a localização de uma frota para atender à demanda dos jogadores, mas evita custos excessivos de hospedagem, como durante um ataque de DDOS. Configure um [alarme do Amazon CloudWatch](#) para avisar quando a capacidade se aproxima do limite máximo, para que você possa avaliar a situação e ajustar manualmente, conforme necessário. (Também é possível [criar um alarme de cobrança](#) para monitorar os custos da AWS.) O limite mínimo é útil para manter a disponibilidade da hospedagem, mesmo quando a demanda dos jogadores é baixa.

Defina os limites de capacidade para as localizações de uma frota no [console do Amazon GameLift](#) ou usando o AWS Command Line Interface (AWS CLI).

Como definir limites de capacidade

Console

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, escolha Hospedagem, Frotas.
3. Na página Frotas, escolha o nome de uma frota ativa para abrir a página de detalhes da frota.
4. Na guia Escalabilidade, em Capacidade de escalabilidade, selecione um local da frota e selecione Editar.
5. Na caixa de diálogo Editar capacidade de escalabilidade, defina as contagens de instâncias para Tamanho mínimo, Instâncias desejadas e Tamanho máximo.
6. Selecione a opção Confirmar.

AWS CLI

1. Verifique as configurações atuais de capacidade. Em uma janela de linha de comando, use o comando [describe-fleet-location-capacity](#) com o ID da frota e o local para o qual você deseja mudar a capacidade. Esse comando retorna um objeto [FleetCapacity](#), que inclui as configurações de capacidade atuais do local. Determine se a nova instância limite poderá acomodar a configuração atual de instâncias desejadas.

```
aws gamelift describe-fleet-location-capacity \
```

```
--fleet-id <fleet identifier> \  
--location <location name>
```

2. Atualize as configurações de limite. Em uma janela da linha de comando, use o comando [update-fleet-capacity](#) com os parâmetros a seguir. Você pode ajustar os limites de instâncias desejadas e contagem de instâncias com o mesmo comando.

```
--fleet-id <fleet identifier>  
--location <location name>  
--max-size <maximum capacity for scaling>  
--min-size <minimum capacity for scaling>  
--desired-instances <fleet capacity goal>
```

Exemplo:

```
aws gamelift update-fleet-capacity \  
  --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \  
  --location us-west-2 \  
  --max-size 10 \  
  --min-size 1 \  
  --desired-instances 10
```

Se sua solicitação for bem-sucedida, o Amazon GameLift retornará o ID da frota. Se o novo valor `max-size` ou `min-size` entrarem em conflito com a `desired-instances` configuração atual, o Amazon GameLift retornará um erro.

Defina manualmente a capacidade de uma frota do Amazon GameLift

Ao criar uma nova frota, o Amazon GameLift define automaticamente as instâncias desejadas para uma instância em cada localização da frota. Em seguida, o Amazon GameLift implanta uma nova instância em cada local. Para alterar a capacidade da frota, adicione uma política de ajuste de escala automático com base em destino ou defina manualmente o número de instâncias que deseja para um local. Para obter mais informações, consulte [Escalar a capacidade de frota](#).

Definir a capacidade de uma frota manualmente pode ser útil ao não precisar de ajuste de escala automático ou quando precisar manter a capacidade em um nível específico. A configuração manual da capacidade só funciona caso não esteja usando uma política de ajuste de escala automático com base em destino. Caso tenha uma política de ajuste de escala automático com base em

destino, ela redefinirá imediatamente a capacidade desejada com base em suas próprias regras de escalabilidade.

É possível definir manualmente a capacidade no console do Amazon GameLift ou usando a AWS Command Line Interface (AWS CLI). O status da frota deve ser ativo.

Suspender o ajuste de escala automático

Suspenda todas as atividades de ajuste de escala automático para cada localização da frota. Com a suspensão do ajuste de escala automático, o número desejado de instâncias no local da frota permanece o mesmo, a menos que seja alterado manualmente. Ao suspender o ajuste de escala automático de um local, isso afeta as políticas atuais da frota e quaisquer políticas que possa ser definida no futuro.

Como definir manualmente capacidade de frota

Console

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, selecione Hospedagem, Frotas.
3. Na página Frotas, escolha o nome de uma frota ativa para abrir a página de detalhes da frota.
4. Na guia Escalabilidade, em Locais de ajuste de escala automático suspensos, selecione cada local para o qual você deseja suspender o ajuste de escala automático e, em seguida, escolha Suspendar.
5. Em Capacidade de escalabilidade, selecione um local a ser definido manualmente e escolha Editar.
6. Na caixa de diálogo Editar capacidade de escalabilidade, defina seu valor preferido para as Instâncias desejadas e escolha Confirmar. Isso informa ao Amazon GameLift o número de instâncias a serem mantidas em um estado ativo, preparadas para hospedar sessões de jogos.

O Amazon GameLift responde às alterações ao implantar instâncias adicionais ou a desligar as desnecessárias. À medida que o Amazon GameLift conclui esse processo, o número de instâncias ativas no local muda para corresponder ao valor atualizado das instâncias desejadas. Esse processo pode demorar um pouco.

AWS CLI

1. Verifique as configurações atuais de capacidade. Em uma janela de linha de comando, use o comando [describe-fleet-location-capacity](#) com o ID da frota e o local para o qual você deseja alterar a capacidade. Esse comando retorna um objeto [FleetCapacity](#), que inclui as configurações de capacidade atuais do local. Determine se os limites de instância podem acomodar a nova configuração desejada de instâncias.

```
aws gamelift describe-fleet-location-capacity \  
  --fleet-id <fleet identifier> \  
  --location <location name>
```

2. Atualizar a capacidade desejada. Use o comando [update-fleet-capacity](#) com o ID da frota, localização e um novo valor para as instâncias desejadas. Se esse valor está fora do intervalo do limite atual, você poderá ajustar os valores limite no mesmo comando.

```
--fleet-id <fleet identifier>  
--location <location name>  
--desired-instances <fleet capacity as an integer>  
--max-size <maximum capacity> [Optional]  
--min-size <minimum capacity> [Optional]
```

Exemplo:

```
aws gamelift update-fleet-capacity \  
  --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \  
  --location us-west-2 \  
  --desired-instances 5 \  
  --max-size 10 \  
  --min-size 1
```

Se sua solicitação for bem-sucedida, o Amazon GameLift retornará o ID da frota. Se a nova configuração de instâncias desejada estiver fora dos limites mínimo e máximo, o Amazon GameLift retornará um erro.

Capacidade de frota em ajuste de escala automático com o Amazon GameLift

Use o ajuste de escala automático no Amazon GameLift para escalar de forma dinâmica a capacidade de frota em resposta à atividade de servidor de jogo. À medida que os jogadores chegam e iniciam as sessões de jogos, o ajuste de escala automático pode adicionar mais instâncias; à medida que a demanda dos jogadores diminui, o ajuste de escala automático pode finalizar instâncias desnecessárias. O ajuste de escala automático é uma forma eficiente para minimizar os recursos de hospedagem e custos, enquanto ainda fornece uma experiência de jogador suave e rápida.

Para usar o ajuste de escala automático, você cria políticas de escalabilidade que informam ao Amazon GameLift quando aumentar ou diminuir a escala. Há dois tipos de políticas de escalabilidade: com base no destino e com base nas regras. A abordagem baseada em destino (rastreamento de destino) é uma solução completa. Nós o recomendamos como a opção mais simples e eficaz. Políticas de escalabilidade com base em regras, que exigem que você defina cada aspecto do processo de realizar um ajuste de escala automático, que é útil para lidar com questões específicas. Esta solução funciona melhor como um suplemento para o ajuste de escala automático com base em destino.

Gerencie o ajuste de escala automático baseado em destino usando o console do Amazon GameLift, o AWS Command Line Interface (AWS CLI), ou um SDK da AWS. Gerencie o ajuste de escala automático baseado em regras usando apenas a AWS CLI um SDK da AWS, apesar de que você possa visualizar políticas de escalabilidade baseada em regras no console.

Tópicos

- [Ajuste de escala automático baseado em destino](#)
- [Ajuste de escala automático com políticas com base nas regras](#)

Ajuste de escala automático baseado em destino

O ajuste de escala automático baseado em destino para o Amazon GameLift ajusta os níveis de capacidade com base na métrica da frota `PercentAvailableGameSessions`. Esta métrica representa o buffer disponível da frota para aumentos repentinos na demanda do jogador.

O principal motivo para manter um buffer de capacidade é o tempo de espera do jogador. Quando slots de sessão de jogo estão prontos e em espera, leva segundos para obter novos jogadores em

sessões de jogo. Se nenhum recurso estiver disponível, os jogadores devem esperar por sessões de jogo existentes serem encerradas ou novos recursos se tornarem disponíveis. Pode levar minutos para iniciar novas instâncias e processos de servidor.

Ao configurar o ajuste de escala automático baseado em destino, especifique o tamanho do buffer que você deseja que a frota mantenha. Como `PercentAvailableGameSessions` mede a porcentagem de recursos disponíveis, o tamanho real do buffer é uma porcentagem da capacidade total da frota. O Amazon GameLift adiciona ou remove instâncias para manter o tamanho do buffer de destino. Com um buffer grande, você minimiza o tempo de espera, mas também paga pelos recursos adicionais que talvez não use. Se os seus jogadores forem mais tolerantes com tempos de espera, você pode reduzir os custos definindo um buffer pequeno.

Para definir o ajuste de escala automático com base em destino

Console

1. Abra o [console do Amazon GameLift](#).
2. No painel de navegação, selecione Hospedagem, Frotas.
3. Na página Frotas, escolha o nome de uma frota ativa para abrir a página de detalhes da frota.
4. Escolha a guia Escalabilidade. Essa guia exibe as métricas de escalabilidade da frota e contém controles históricos para ajustar as configurações de escalabilidade atuais.
5. Em Capacidade de escalabilidade, verifique se os limites de tamanho mínimo e máximo são apropriados para a frota. Com o ajuste de escala automático ativado, a capacidade se ajusta entre esses dois limites.
6. Em Política de ajuste de escala automático baseada em destino, selecione Editar.
7. Na caixa de diálogo Editar política de ajuste de escala automático baseada em destino, em Porcentagem de sessões de jogo disponíveis, defina a porcentagem que você deseja manter e escolha Confirmar. Depois de confirmar as configurações, o Amazon GameLift adiciona uma nova política baseada em destino em Política de ajuste de escala automático baseada em destino.

AWS CLI

1. Defina limites de capacidade. Defina os valores de limite usando o comando [update-fleet-capacity](#). Para obter mais informações, consulte [Defina os limites de capacidade do Amazon GameLift](#).

2. Crie uma política. Abra uma janela da linha de comando e use o comando [put-scaling-policy](#) com suas configurações de parâmetro de política. Para atualizar uma política existente, especifique o nome da política e forneça uma versão completa da política atualizada.

```
--fleet-id <unique fleet identifier>
--name "<unique policy name>"
--policy-type <target- or rule-based policy>
--metric-name <name of metric>
--target-configuration <buffer size>
```

Exemplo:

```
aws gamelift put-scaling-policy \
  --fleet-id "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa" \
  --name "My_Target_Policy_1" \
  --policy-type "TargetBased" \
  --metric-name "PercentAvailableGameSessions" \
  --target-configuration "TargetValue=5"
```

Ajuste de escala automático com políticas com base nas regras

Regra de escalabilidade com base em políticas no Amazon GameLift fornecem controle refinado ao ajuste de escala automático da capacidade de uma frota em resposta à atividade do jogador. Para cada política, é possível vincular a escalabilidade a uma das várias métricas da frota, identificar uma variação e personalizar a resposta de expansão ou evento de redução. As políticas com base em regras são úteis para completar a [escalabilidade com base em destino](#) para lidar com circunstâncias especiais.

Uma política baseada em regras afirma o seguinte: "Se a métrica de uma frota atingir ou ultrapassar um valor limite para um determinado período, altere a capacidade da frota em uma determinada quantidade." Este tópico descreve a sintaxe usada para estruturar uma declaração de política e fornece ajuda com a criação e o gerenciamento de políticas com base em regras.

Gerenciar políticas baseadas em regras

Crie, atualize ou exclua políticas com base em regras usando um SDK da AWS ou o AWS Command Line Interface (AWS CLI) com a [API do serviço do Amazon GameLift](#). É possível visualizar todas as políticas ativas no console do Amazon GameLift.

Para interromper temporariamente todas as políticas de escalabilidade para uma frota, use o comando [stop-fleet-actions](#) da AWS CLI.

Para criar ou atualizar uma política de escalabilidade baseada em regras (AWS CLI):

1. Defina limites de capacidade. Defina um ou ambos os valores de limite usando o comando [update-fleet-capacity](#). Para obter mais informações, consulte [Defina os limites de capacidade do Amazon GameLift](#).
2. Crie uma política. Abra uma janela da linha de comando e use o comando [put-scaling-policy](#) com suas configurações de parâmetro de política. Para atualizar uma política existente, especifique o nome da política e forneça uma versão completa da política atualizada.

```
--fleet-id <unique fleet identifier>
--name "<unique policy name>"
--policy-type <target- or rule-based policy>
--metric-name <name of metric>
--comparison-operator <comparison operator>
--threshold <threshold integer value>
--evaluation-periods <number of minutes>
--scaling-adjustment-type <adjustment type>
--scaling-adjustment <adjustment amount>
```

Exemplo:

```
aws gamelift put-scaling-policy \
  --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \
  --name "Scale up when AGS<50" \
  --policy-type RuleBased \
  --metric-name AvailableGameSessions \
  --comparison-operator LessThanThreshold \
  --threshold 50 \
  --evaluation-periods 10 \
  --scaling-adjustment-type ChangeInCapacity \
  --scaling-adjustment 1
```

Para excluir uma regra de escalabilidade com base em política usando a AWS CLI:

- Abra uma janela de linha de comando e use o comando [delete-scaling-policy](#) com o ID da frota e o nome da política.

Exemplo:

```
aws gamelift delete-scaling-policy \  
  --fleet-id fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa \  
  --name "Scale up when AGS<50"
```

Regras de sintaxe para o ajuste de escala automático

Para construir uma declaração de política de escalabilidade baseada em regra, especifique seis variáveis:

Se *<nome da métrica>* permanecer *<operador de comparação>* *<valor limite>* para *<período de avaliação>*, altere a capacidade da frota usando *<tipo de ajuste>* para/por *<valor de ajuste>*.

Por exemplo, esta declaração de política inicia um evento de expansão sempre que uma frota de capacidade extra é menor do que o que é necessário para lidar com 50 novas sessões de jogos:

Se AvailableGameSessions permanecer em less than 50 para 10 minutes, altere a capacidade da frota usando ChangeInCapacity por 1 instances.

Nome da métrica

Para iniciar um evento de escalabilidade, vincule uma política de ajuste de escala automático para uma das seguintes métricas específicas da frota. Para obter descrições de métrica completa, consulte [Métricas do Amazon GameLift para frotas](#).

- Ativar sessões de jogos
- Sessões de jogos ativas
- Sessões de jogos disponíveis
- Sessões de jogos disponíveis em porcentagem
- Instâncias ativas
- Sessões de jogadores disponíveis
- Sessões de jogadores atuais
- Instâncias ociosas
- Instâncias ociosas em porcentagem

Caso a frota esteja em uma fila de sessão de jogo, use as seguintes métricas:

- Profundidade da fila – O número de solicitações de sessão de jogo pendentes nesta frota é o melhor local de hospedagem disponível.
- Tempo de espera – Tempo de espera específico da frota. O período de tempo em que a solicitação de sessão de jogo pendente mais antigo está esperando para ser atendido. O tempo de espera da frota é igual ao tempo da solicitação atual mais antiga na fila.

Operador de comparação

Informa ao Amazon GameLift como comparar os dados da métrica com o valor limite. Entre os operadores de comparação válidos estão maior que (>), menor que (<), maior que ou igual a (>=) e menor que ou igual (<=).

Valor limite

Quando o valor de métrica especificado encontra ou cruza o valor de limite, ele pode iniciar um evento de escalabilidade. Esse valor é sempre um inteiro positivo.

Período de avaliação

A métrica deve atingir ou ultrapassar o valor limite em toda a duração do período de avaliação antes de iniciar um evento de escalabilidade. O tamanho do período de avaliação é consecutivo; se a métrica não atingir o limite, o período de avaliação recomeçará.

Tipo de ajuste e valor

Este conjunto de variáveis funciona agrupado para especificar como você deseja que o Amazon GameLift ajuste a capacidade da frota quando essa política inicia um evento de escalabilidade.

Escolha um dos três tipos de ajustes possíveis:

- Alteração na capacidade – Aumente ou diminua a capacidade atual segundo o número especificado de instâncias. Defina o valor de ajuste para o número de instâncias a serem adicionadas a ou subtraídas da frota. Os valores positivos adicionam instâncias, enquanto os valores negativos removem instâncias. Por exemplo, um valor de "-10" reduz a frota por 10 instâncias, independentemente do tamanho total da frota.
- Alteração percentual na capacidade – Aumente ou diminua a capacidade atual segundo uma porcentagem especificada. Defina o valor de ajuste para a porcentagem na qual você deseja aumentar ou diminuir a capacidade da frota. Os valores positivos adicionam instâncias, enquanto os valores negativos removem instâncias. Por exemplo, para uma frota com 50 instâncias, uma alteração de porcentagem de "20" adiciona 10 instâncias à frota.
- Capacidade exata — Aumentar ou diminuir a capacidade atual para um valor específico. Defina o valor de ajuste para o número exato de instâncias que você deseja manter na frota.

Dicas para o ajuste de escala automático com base em regras

As sugestões a seguir podem ajudar a obter o máximo de ajuste de escala automático com políticas baseadas em regras.

Usar várias políticas

Você pode ter várias políticas de ajuste de escala automático para uma frota ao mesmo tempo. O cenário mais comum é ter uma política baseada em destino gerenciar a maioria dos requisitos de escalabilidade e usar políticas com base em regras para lidar com problemas em parâmetros. Não há limites no uso de várias políticas.

Com várias políticas, cada uma delas se comporta de forma independente. Não há como controlar a sequência de eventos de escalabilidade. Por exemplo, se você tiver várias políticas aumentando a escalabilidade, será possível que a atividade do jogador inicie vários eventos de escalabilidade simultaneamente. Evite políticas que se iniciem mutuamente. Por exemplo, você poderá criar um loop infinito se aumentar e diminuir políticas que definam a capacidade além do limite uma da outra.

Definir as capacidades máxima e mínima

Cada frota tem um limite de capacidade máxima e mínima. Esse recurso é especialmente importante quando se usa o ajuste de escala automático. O ajuste de escala automático jamais define a capacidade como um valor fora desse intervalo. Por padrão, frotas recém-criadas têm um mínimo de 0 e um máximo de 1. Para a política de ajuste de escala automático afetar a capacidade conforme desejado, aumente o valor máximo.

A capacidade da frota também é restringida por limites no tipo de instância da frota e pelas cotas de serviço na Conta da AWS. Não é possível definir um mínimo e um máximo fora desses limites e cotas de conta

Acompanhar métricas depois de uma alteração na capacidade

Depois de alterar a capacidade em resposta a uma política de ajuste de escala automático, o Amazon GameLift aguardará 10 minutos até responder aos gatilhos dessa política. Essa espera dá ao Amazon GameLift tempo para adicionar as novas instâncias, iniciar os servidores de jogos, conectar jogadores e iniciar a coleta de dados das novas instâncias. Durante esse período, o Amazon GameLift avalia a política em relação à métrica e rastreia o período de avaliação da política, reiniciado após a ocorrência de um evento de escalabilidade. Isso significa que uma política de escalabilidade poderá iniciar outro evento de escalabilidade logo depois que o tempo de espera terminar.

Não há tempo de espera entre os eventos de escalabilidade que iniciam diferentes políticas de ajuste de escala automático.

Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo

Uma fila de sessões de jogo é o principal mecanismo para processar novas solicitações de sessões de jogo e localizar servidores de jogos disponíveis para hospedá-las. As filas oferecem benefícios significativos para desenvolvedores e jogadores de jogos. Eles incluem:

- As filas oferecem a melhor posicionamento possível. Ao processar solicitações de posicionamento de sessões de jogo, uma fila usa algoritmos do Amazon GameLift para priorizar os locais das filas com base em um conjunto de preferências definidas.
- Hospede jogos em frotas spot com preços mais baixos. Use as filas para otimizar o uso das frotas spot AWS, que oferecem custos de hospedagem significativamente mais baixos. Por padrão, as filas sempre tentam posicionar novas sessões de jogo nas frotas spot.
- Coloque novos jogos com mais rapidez durante uma alta demanda. As filas usam vários locais possíveis para posicionamentos. Isso significa que sempre há capacidade alternativa caso o local de posicionamento preferencial não esteja disponível.
- Torne a disponibilidade de jogos mais resiliente. Interrupções podem acontecer. Ao usar uma fila de vários locais, evite lentidão ou interrupções no acesso dos jogadores ao seu jogo.
- Use a capacidade extra da frota de forma mais eficiente. Para lidar com surtos inesperados na demanda de jogadores, as filas oferecem acesso rápido à capacidade extra de hospedagem. Os locais da frota em uma fila oferecem capacidade de backup um ao outro. Os locais aumentam ou diminuem com base na demanda dos jogadores.
- Obtenha métricas sobre posicionamentos de sessões de jogo e desempenho de fila. O Amazon GameLift emite métricas da fila, incluindo estatísticas sobre sucessos e falhas de posicionamento, quantidade de solicitações na fila e tempo médio que as solicitações permanecem na fila. Você pode visualizar essas métricas às filas no console do Amazon GameLift ou no CloudWatch.

Para começar a usar as filas, consulte [Criar uma fila de sessões de jogo](#).

Criar uma fila de sessões de jogo

Este tópico descreve como criar uma fila que ofereça uma experiência ao jogador com latência mínima e que use com eficiência os recursos de hospedagem. Para obter mais informações sobre

filas de sessões de jogo e como elas funcionam, consulte [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).

Estes atributos do Amazon GameLift exigem filas:

- [Como marcar jogos com o FlexMatch](#)
- [Use instâncias spot com a Amazon GameLift](#)

Defina o escopo da sua fila

A população de jogadores do seu jogo pode ter grupos de jogadores que não deveriam jogar juntos. Por exemplo, se você publicar seu jogo em dois idiomas, cada idioma deverá ter seus próprios servidores de jogos.

Para configurar o posicionamento de sessões de jogo para sua população de jogadores, crie uma fila separada para cada segmento de jogadores. Avalie cada fila para posicionar os jogadores nos servidores de jogos corretos. Algumas formas comuns de definir o escopo das filas incluem:


- Por localizações geográficas. Ao implantar seus servidores de jogos em várias áreas geográficas, você pode criar filas para jogadores em cada local a fim de reduzir a latência dos jogadores.
- Por variações de compilação ou script. Se tiver mais de uma variação do seu servidor de jogos, talvez você esteja apoiando grupos de jogadores que não podem jogar nas mesmas sessões de jogo. Por exemplo, compilações ou scripts de servidores de jogos podem oferecer suporte a diferentes idiomas ou tipos de dispositivos.
- Por tipos de eventos. Você pode criar uma fila especial para gerenciar jogos para participantes de torneios ou outros eventos especiais.

Crie uma política de latência do jogador

Se suas solicitações de posicionamento incluírem dados de latência do jogador, o algoritmo descobrir sessões de jogo em locais com a menor latência média para todos os jogadores. O posicionamento de sessões de jogo com base na latência média do jogador impede que o Amazon GameLift coloque a maioria dos jogadores em jogos com alta latência. No entanto, o Amazon GameLift ainda coloca jogadores com latência extrema. Para acomodar esses jogadores, crie políticas de latência do jogador.

Uma política de latência do jogador impede que o Amazon GameLift coloque uma sessão de jogo solicitada em qualquer lugar em que os jogadores da solicitação experimentem latência acima do

valor máximo. As políticas de latência do jogador também podem impedir que o Amazon GameLift combine solicitações de sessões de jogo com jogadores de maior latência.

 Tip

Para gerenciar regras específicas de latência, como exigir latência semelhante em todos os jogadores de um grupo, você pode usar o [Amazon GameLift FlexMatch](#) para criar regras de marcação de jogos com base na latência.

Por exemplo, considere essa fila com um tempo limite de cinco minutos e as seguintes políticas de latência do jogador:

1. Passe 120 segundos procurando um local onde todas as latências do jogador sejam inferiores a 50 milissegundos.
2. Passe 120 segundos procurando um local onde todas as latências do jogador sejam inferiores a 100 milissegundos.
3. Passe o tempo restante da fila até o tempo limite procurando um local onde todas as latências do jogador sejam inferiores a 200 milissegundos.

Create queue

Queue settings

Name

The name must be unique and have 1-128 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen).

Timeout

Specify how long GameLift tries to place a game session before stopping.

 seconds

Must be 10-600 seconds.

 We recommend setting player latency policies, unless you're using GameLift FlexMatch.



Player latency policies - *optional*

Add policies to help place players into games with lower latency. Use multiple policies to reduce latency requirements per policy so that each player eventually finds a match.

0 seconds left to allocate

100%

Period start

Seconds

Period end

Seconds

Max player latency

Milliseconds

Remove

Seconds

Seconds

Milliseconds

Remove

Seconds

Seconds

Milliseconds

Remove

Add policy

Criar uma fila de vários locais

Recomendamos um design de vários locais para todas as filas. Esse design pode melhorar a velocidade do posicionamento e a resiliência da hospedagem. É necessário um design de vários locais para usar os dados de latência do jogador e posicionar os jogadores em sessões de jogo com

latência mínima. Se estiver criando filas de vários locais que usam frotas de instâncias spot, siga as instruções em [Tutorial: Configurar uma fila de sessão de jogo para instâncias spot](#).

Uma forma de criar uma fila de vários locais é adicionar uma [frota de vários locais](#) a uma fila. Dessa forma, a fila pode posicionar sessões de jogo em qualquer um dos locais da frota. Você também pode adicionar outras frotas com configurações ou locais residenciais diferentes para redundância. Se você estiver usando uma frota de instâncias spot de vários locais, siga as melhores práticas e inclua uma frota de instância sob demanda com os mesmos locais.

O exemplo a seguir descreve o processo de criação de uma fila básica de vários locais. Neste exemplo, usamos duas frotas: uma frota de instâncias spot e uma frota de instâncias sob demanda. Cada frota tem o seguinte Regiões da AWS para locais de posicionamento: us-east-1, us-east-2, ca-central-1 e us-west-2.

Para criar uma fila básica de vários locais com frotas de vários locais

1. Escolha um local para criar a fila. Você pode minimizar a latência da solicitação colocando a fila em um local próximo ao local onde você implantou o serviço ao cliente. Neste exemplo, criamos a fila em us-east-1.
2. Crie uma nova fila e adicione suas frotas de vários locais como destinos de fila. A ordem dos destinos determina como o Amazon GameLift coloca as sessões de jogo. Neste exemplo, listamos a frota de instâncias spot primeiro e a frota de instâncias sob demanda em segundo lugar.
3. Defina a ordem de prioridade de posicionamento de sessões de jogo da fila. Essa ordem determina onde a fila procura primeiro por um servidor de jogos disponível. Neste exemplo, usamos a ordem de prioridade padrão.
4. Defina a ordem dos locais. Se você não definir a ordem dos locais, o Amazon GameLift usará os locais em ordem alfabética.

Game session placement locations

Locations where the queue can place new game sessions.

Locations

Choose locations

ca-central-1 ✕
Canada (Central)

us-west-2 ✕
US West (Oregon)

us-east-2 ✕
US East (Ohio)

us-east-1 ✕
US East (N. Virginia)

Destination order

An ordered list of fleets and aliases that the queue can use for game session placement.

	Region	Type	Name	
⋮	us-east-1 ▼	Fleet ▼	TestFleet-SPOT ▼	Remove
⋮	us-east-1 ▼	Fleet ▼	TestFleet-ONDEMAND ▼	Remove

Add Destination

Game session placement priority

The values that GameLift uses to prioritize game session placement. The default order is latency, cost, destination, and location.

- Latency**
Prioritize locations with the lowest average player latency.
- Cost**
Prioritize destinations with the lowest current hosting cost.
- Destination**
Prioritize based on the defined destination order.
- Location**
Prioritize based on the defined location order.

▼ Location order

An ordered list of locations that the queue can use for game session placement.

Location	
ca-central-1	<input type="button" value="Remove"/>
us-east-1	<input type="button" value="Remove"/>
us-east-2	<input type="button" value="Remove"/>
us-west-2	<input type="button" value="Remove"/>

Priorize o posicionamento de sessões de jogo

O Amazon GameLift usa o algoritmo FleetIQ para determinar onde posicionar uma nova sessão de jogo com base em um conjunto ordenado de critérios. Você pode usar a ordem de prioridades padrão ou personalizá-la.

Ordem de prioridades padrão

Para solicitações de posicionamento que incluam dados de latência do jogador, o FleetIQ prioriza os critérios de posicionamento de sessões de jogo na seguinte ordem padrão:

1. Latência: latência média mais baixa para todos os jogadores da solicitação.
2. Custo: menor custo de hospedagem, se a latência for igual em vários locais. O custo de hospedagem é baseado, principalmente, em uma combinação do tipo e local da instância.
3. Destino: ordem dos destinos, se a latência e o custo forem iguais em vários locais. O FleetIQ prioriza os destinos com base na ordem listada na configuração da fila.
4. Local: ordem dos locais, se a latência, o custo e o destino forem iguais em vários locais. O FleetIQ prioriza os locais com base na ordem listada na configuração da fila.

Ordem de prioridades personalizada

Para personalizar a ordem de prioridades de uma fila no console [Amazon GameLift](#), arraste o valor da prioridade até a posição em que você deseja que ela fique. Para personalizar a ordem de prioridades de uma fila usando o AWS Command Line Interface (AWS CLI), use o comando [create-game-session-queue](#) com a opção `--priority-configuration`. Você pode usar esse comando para criar uma nova fila ou atualizar uma fila existente.

O algoritmo FleetIQ anexa quaisquer critérios não explicitamente mencionados ao final da sua lista, com base na ordem padrão. Se incluir o critério de localização na sua configuração de prioridades, você também deverá fornecer uma lista ordenada de locais.

Crie várias filas conforme necessário

Dependendo do jogo e dos jogadores, talvez você queira criar mais de uma fila de sessões de jogo. Quando o serviço do cliente do jogo solicita uma nova sessão de jogo, ele especifica que fila da sessão de jogo usar. Para ajudá-lo a determinar se você deve usar várias filas, considere:

- Variações do servidor de jogos. Você pode criar uma fila separada para cada variação do servidor de jogos. Todas as frotas de uma fila devem implantar servidores de jogos compatíveis. Isso ocorre porque os jogadores que usam a fila para entrar nos jogos devem poder jogar em qualquer um dos servidores de jogos da fila.
- Grupos de jogadores diferentes. Você pode personalizar a forma como o Amazon GameLift coloca as sessões de jogo com base no grupo de jogadores. Por exemplo, você pode precisar de filas personalizadas para determinados modos de jogo que exigem um tipo especial de instância ou de configuração de runtime. Ou, talvez, você queira uma fila especial para gerenciar posicionamentos em um torneio ou outro evento.

- Métricas da fila de sessões de jogo. Você pode configurar filas com base em como deseja coletar as métricas de posicionamento de sessões de jogo. Para obter mais informações, consulte [Métricas do Amazon GameLift para filas](#).

Avaliar métricas de fila

Use métricas para avaliar o desempenho das suas filas. Você pode visualizar métricas relacionadas às filas no [console do Amazon GameLift](#) ou no Amazon CloudWatch. Para obter uma lista e descrições de métricas de fila, consulte [Métricas do Amazon GameLift para filas](#).

As métricas de fila podem fornecer informações sobre o seguinte:

- Desempenho geral da fila: as métricas da fila indicam com que sucesso uma fila responde às solicitações de posicionamento. Essas métricas também podem ajudá-lo a identificar quando e por que os posicionamentos falham. Para filas com frotas escaladas manualmente, as métricas `AverageWaitTime` e `QueueDepth` podem indicar quando você deve ajustar a capacidade de uma fila.
- Desempenho do algoritmo FleetIQ: para solicitações de posicionamento usando o algoritmo FleetIQ, as métricas mostram com que frequência o algoritmo descobre o posicionamento ideal de sessões de jogo. O posicionamento pode priorizar o uso de recursos com a menor latência do jogador ou recursos com o menor custo. Também existem métricas de erro que identificam motivos comuns pelos quais o Amazon GameLift não consegue descobrir um posicionamento ideal. Para obter mais informações sobre métricas, consulte [Monitorar o Amazon GameLift com o Amazon CloudWatch](#).
- Posicionamentos específicos de locais: para filas de vários locais, as métricas mostram posicionamentos bem-sucedidos por local. Com filas que usam o algoritmo FleetIQ, esses dados fornecem informações úteis sobre onde a atividade do jogador ocorre.

Ao avaliar métricas de desempenho do algoritmo FleetIQ, considere as seguintes sugestões:

- Para rastrear a taxa da fila de descobertas de um posicionamento ideal, use a métrica `PlacementsSucceeded` em combinação com as métricas do FleetIQ para obter a menor latência e o menor preço.
- Para aumentar a taxa da fila de descobertas de um posicionamento ideal, analise as seguintes métricas de erro:

- Se `FirstChoiceOutOfCapacity` for alto, ajuste o escalonamento da capacidade para as frotas da fila.
- Se a métrica `FirstChoiceNotViable` de erro for alta, veja suas frotas de instâncias Spot. As frotas de instâncias spot são consideradas inviáveis quando a taxa de interrupção de um determinado tipo de instância é muito alta. Para resolver esse problema, altere a fila de modo que ela use as frotas de instâncias spot com diferentes tipos de instância. Recomendamos que você inclua frotas de instâncias spot com diferentes tipos de instância em cada local.

Melhores práticas do Amazon GameLift em filas de sessão de jogo

Aqui estão algumas melhores práticas que podem ajudá-lo a criar filas de sessões de jogo eficazes para o posicionamento das sessões de jogo.

Melhores práticas para filas com qualquer tipo de frota

Uma fila contém uma lista de destinos de frota onde novas sessões de jogo podem ser colocadas. Cada frota pode ter instâncias implantadas em várias localizações geográficas. Ao escolher um posicionamento, a fila seleciona uma combinação de uma frota e uma localização da frota. Você fornece um conjunto de prioridades para a fila usar ao escolher um posicionamento.

Considere as seguintes diretrizes e melhores práticas:

- Adicione frotas em locais que cubram seus jogadores. Você pode adicionar frotas e aliases em qualquer local disponível. A localização é importante se você estiver fazendo posicionamentos com base na latência relatada do jogador.
- Use aliases para todas as frotas. Atribua um alias a cada frota em uma fila e use os nomes de alias ao definir destinos em sua fila.
- Use a mesma versão ou script de jogo ou similar para todas as frotas. A fila pode colocar os jogadores em sessões de jogo em qualquer frota na fila. Os jogadores devem poder jogar em qualquer sessão de jogo em qualquer frota.
- Crie frotas em pelo menos dois locais. Ao ter servidores de jogos hospedados em pelo menos um outro local, você atenua o impacto das interrupções regionais em seus jogadores. Você poderá reduzir a escala de suas frotas de backup e usar o ajuste de escala automático para aumentar a capacidade se o uso aumentar.
- Priorize o posicionamento da sessão de jogo. Uma fila prioriza as opções de posicionamento com base em vários elementos, incluindo a ordem da lista de destinos.

- Crie sua fila no mesmo local do seu atendimento ao cliente. Ao colocar sua fila em um local próximo ao atendimento ao cliente, você pode minimizar a latência de comunicação.
- Use frotas com vários locais. Use a configuração do filtro de fila para evitar que a fila coloque sessões de jogo em locais específicos. Você poderá usar pelo menos duas frotas de vários locais com residências diferentes para mitigar o impacto das colocações de jogos durante uma interrupção regional.
- Use a mesma configuração de certificado TLS para todas as frotas. Os clientes de jogos que se conectam às sessões de jogo em suas frotas devem ter protocolos de comunicação compatíveis.

Melhores práticas para filas com frotas Spot

Se sua fila incluir frotas Spot, configure uma fila resiliente. Isso aproveita a economia de custos com as frotas Spot e, ao mesmo tempo, minimiza o efeito das interrupções nas sessões de jogo. Para obter ajuda sobre como criar frotas e filas de sessões de jogo corretamente para uso com frotas Spot, consulte [Tutorial: Configurar uma fila de sessão de jogo para instâncias spot](#). Para obter mais informações sobre instâncias spot, consulte [Use instâncias spot com a Amazon GameLift](#).

Além das melhores práticas gerais na seção anterior, considere estas melhores práticas específicas do SPOT:

- Crie pelo menos uma frota sob demanda em cada local. As frotas sob demanda fornecem servidores de jogos de backup para seus jogadores. Você poderá reduzir a escala de suas frotas de backup até que sejam necessárias e usar o ajuste de escala automático para aumentar a capacidade sob demanda quando as frotas spot não estiverem disponíveis.
- Selecione diferentes tipos de instância em várias frotas Spot em um local. Se um tipo de Instância Spot ficar temporariamente indisponível, a interrupção afetará somente uma frota Spot no local. A melhor prática é escolher tipos de instância amplamente disponíveis e usar tipos de instância da mesma família (por exemplo, m5.large, m5.xlarge, m5.2xlarge). Use o [console do Amazon GameLift](#) para visualizar dados históricos de preços para tipos de instância.

Criar uma fila de sessões de jogo

As filas são usadas para posicionar novas sessões de jogo com os melhores recursos de hospedagem disponíveis em várias frotas e regiões. Para saber mais sobre como criar filas para o seu jogo, consulte [Criar uma fila de sessões de jogo](#).

Em um cliente de jogo, novas sessões de jogo são iniciadas com filas usando solicitações de posicionamento. Saiba mais sobre o posicionamento de sessões de jogo em [Criar sessões de jogos](#).

Ao atualizar o destino da fila em uma fila, há um curto período de transição (de até 30 segundos) durante o qual as sessões de jogo posicionadas nos destinos da fila ainda podem acabar na frota antiga.

Console

1. No [console do Amazon GameLift](#), na página de navegação, escolha Filas.
2. Na página Filas, escolha Criar fila.
3. Na página Criar fila, em Configurações da fila, faça o seguinte:
 - a. Em Nome, insira um nome de fila.
 - b. Em Tempo limite, digite por quanto tempo você deseja que o Amazon GameLift tente posicionar uma sessão de jogo antes de parar. O Amazon GameLift procura recursos disponíveis em qualquer frota até a solicitação expirar.
 - c. (Opcional) Para Políticas de latência do jogadores, insira por quanto tempo o Amazon GameLift deve procurar recursos dentro da latência máxima definida. Adicione mais políticas para reduzir gradualmente a latência máxima. Para adicionar mais políticas, escolha Adicionar política.
4. Em Locais de posicionamento das sessões de jogo), selecione os locais a serem incluídos na fila. Por padrão, Todos os locais estão incluídos. Todas as frotas na fila devem ter a mesma configuração de certificado. Todas as frotas devem estar executando compilações de jogos compatíveis com os clientes de jogos que usam a fila.
5. Em Ordem de destinos, adicione um ou mais destinos à fila.
 - a. Escolha Adicionar destino.
 - b. Selecione o Local em que o destino está.
 - c. Selecione o tipo para o seu destino.
 - d. Na lista resultante de nomes de frota ou alias, selecione aquele que você deseja adicionar.
 - e. Se você tiver vários destinos, defina a ordem padrão arrastando o ícone de seis pontos à esquerda do destino. O Amazon GameLift usa essa ordem ao pesquisar destinos de recursos disponíveis para posicionar uma nova sessão de jogo.

6. Para Prioridade de posicionamento da sessão de jogo), adicione e arraste os valores Latência, Custo, Destino e Local para definir como o Amazon GameLift prioriza as frotas na sua fila. Para obter mais informações sobre como priorizar frotas, consulte [Priorize o posicionamento de sessões de jogo](#).
7. Adicione locais a Ordem de localização e arraste-os até a prioridade que a fila deve usar. Se Local for a última prioridade para o posicionamento da sessão de jogo, o Amazon GameLift o usará como desempate.
8. (Opcional) Em Configurações de notificação de eventos, faça o seguinte:
 - a. Selecione ou crie um tópico do SNS para receber notificações de eventos relacionadas ao posicionamento. Para obter mais informações sobre notificações de eventos, consulte [Configurar notificações de eventos para o posicionamento de sessões de jogo](#).
 - b. Adicione Dados de eventos personalizados para anexar a eventos criados por essa fila.
9. (Opcional) Adicione Tags. Para obter mais informações sobre marcação, consulte [Marcação de recursos da AWS](#).
10. Escolha Criar.

AWS CLI

Example Criar uma fila

O exemplo a seguir cria uma fila de sessão de jogo com estas configurações:

- Um tempo limite de cinco minutos
- Dois destinos de frota
- Filtros para permitir apenas locais em `us-east-1`, `us-east-2`, `us-west-2` e `ca-central-1`
- Prioriza destinos com base no custo e, em seguida, nos locais na ordem definida.

```
aws gamelift create-game-session-queue \  
  --name "sample-test-queue" \  
  --timeout-in-seconds 300 \  
  --destinations DestinationArn="arn:aws:gamelift:us-east-1:111122223333:fleet/  
fleet-772266ba-8c82-4a6e-b620-a74a62a93ff8" DestinationArn="arn:aws:gamelift:us-  
east-1:111122223333:fleet/fleet-33f28fb6-aa8b-4867-85b4-ceb217bf5994" \  
  --filter-configuration "AllowedLocations=us-east-1, ca-central-1, us-east-2, us-  
west-2" \  

```

```
--priority-configuration
PriorityOrder="LOCATION","DESTINATION",LocationOrder="us-east-1","us-east-2","ca-
central-1","us-west-2" \
--notification-target "arn:aws:sns:us-east-1:111122223333:gamelift-test.fifo"
```

Note

Você pode obter valores de ARN da frota e do chamando [describe-fleet-attributes](#) ou [describe-alias](#) com o ID da frota ou do alias.

Se a solicitação `create-game-session-queue` for bem-sucedida, o Amazon GameLift retornará um objeto [GameSessionQueue](#) com a nova configuração de fila. Já é possível enviar solicitações para a fila usando [StartGameSessionPlacement](#).

Example Criar uma fila com políticas de latência de jogador

O exemplo a seguir cria uma fila de sessões de jogo com estas configurações:

- Um tempo limite de dez minutos
- Três destinos de frota
- Um conjunto de políticas de latência dos jogadores

```
aws gamelift create-game-session-queue \
  --name "matchmaker-queue" \
  --timeout-in-seconds 600 \
  --destinations DestinationArn=arn:aws:gamelift:us-east-1::alias/alias-a1234567-
b8c9-0d1e-2fa3-b45c6d7e8910 \
    DestinationArn=arn:aws:gamelift:us-west-2::alias/alias-b0234567-
c8d9-0e1f-2ab3-c45d6e7f8901 \
    DestinationArn=arn:aws:gamelift:us-west-2::fleet/fleet-f1234567-
b8c9-0d1e-2fa3-b45c6d7e8912 \
  --player-latency-policies
  "MaximumIndividualPlayerLatencyMilliseconds=50,PolicyDurationSeconds=120" \
  "MaximumIndividualPlayerLatencyMilliseconds=100,PolicyDurationSeconds=120" \
  "MaximumIndividualPlayerLatencyMilliseconds=150" \
```

Se a solicitação `create-game-session-queue` for bem-sucedida, o Amazon GameLift retornará um objeto [GameSessionQueue](#) com a nova configuração de fila.

Configurar notificações de eventos para o posicionamento de sessões de jogo

Você pode usar as notificações de eventos para monitorar o status de solicitações individuais de posicionamento. Recomendamos configurar notificações de eventos para todos os jogos com alto volume de atividades de posicionamento.

Há duas opções para a configuração de notificações de eventos.

- Faça com que o Amazon GameLift publique notificações de eventos em um tópico do Amazon Simple Notification Service (Amazon SNS) usando uma fila.
- Use eventos do Amazon EventBridge publicados automaticamente e seu conjunto de ferramentas para gerenciar eventos.

Para obter uma lista de eventos de posicionamento de sessões de jogo emitidos pelo Amazon GameLift, consulte [Eventos de posicionamento de sessões de jogo](#).

Configurar um tópico do SNS

Para que o Amazon GameLift publique todos os eventos gerados por uma fila de sessões de jogo em um tópico, defina o campo de destino da notificação como um tópico.

Para configurar um tópico do SNS para notificação de eventos do Amazon GameLift

1. Faça login no AWS Management Console e abra o console do Amazon SNS em <https://console.aws.amazon.com/sns/v3/home>.
2. Na página Topics do SNS, escolha Create topic (Criar tópico) e siga as instruções para criá-lo.
3. Em Access Policy (Política de acesso), faça o seguinte:
 - a. Escolha o método Advanced (Avançado).
 - b. Adicione a seguinte seção em negrito do objeto JSON à política existente.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
```

```

    "Principal": {
      "AWS": "*"
    },
    "Action": [
      "SNS:GetTopicAttributes",
      "SNS:SetTopicAttributes",
      "SNS:AddPermission",
      "SNS:RemovePermission",
      "SNS:DeleteTopic",
      "SNS:Subscribe",
      "SNS:ListSubscriptionsByTopic",
      "SNS:Publish"
    ],
    "Resource": "arn:aws:sns:your_region:your_account:your_topic_name",
    "Condition": {
      "StringEquals": {
        "AWS:SourceAccount": "your_account"
      }
    }
  },
  {
    "Sid": "__console_pub_0",
    "Effect": "Allow",
    "Principal": {
      "Service": "gamelift.amazonaws.com"
    },
    "Action": "sns:Publish",
    "Resource": "arn:aws:sns:your_region:your_account:your_topic_name",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn":
          "arn:aws:gamelift:your_region:your_account:gamesessionqueue/your_queue_name"
      }
    }
  }
]
}

```

- c. (Opcional) Adicione mais controle de acesso ao tópico incluindo condições à política de recursos.
4. Escolha Create topic (Criar tópico).
5. Depois de criar seu tópico do SNS, adicione-o às filas durante a criação das filas ou edite uma fila existente para adicioná-lo.

Configurar um tópico do SNS com a criptografia do lado do servidor

Com a criptografia do lado do servidor (SSE), é possível armazenar dados confidenciais em tópicos criptografados. A SSE protege o conteúdo de mensagens nos tópicos do Amazon SNS usando chaves gerenciadas no AWS Key Management Service (AWS KMS). Para obter mais informações sobre criptografia no lado do servidor com o Amazon SNS, consulte [Criptografia em repouso](#) no Guia do desenvolvedor do Amazon Simple Notification Service.

Para configurar um tópico do SNS com a criptografia do lado do servidor, consulte os tópicos a seguir:

- [Criar chaves](#) no Guia do desenvolvedor do AWS Key Management Service
- [Habilitar a SSE para um tópico](#) no Guia do desenvolvedor do Amazon Simple Notification Service

Ao criar a chave do KMS, use a seguinte política de chave do KMS:

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

Configurar o EventBridge

O Amazon GameLift publica automaticamente todos os eventos de posicionamento das sessões de jogo no EventBridge. Com o EventBridge, é possível configurar regras para que eventos sejam roteados para destinos para processamento. Por exemplo, é possível definir uma regra para rotear o evento `PlacementFulfilled` para uma função AWS Lambda que gerencia tarefas que precedem a conexão com uma sessão de jogo. Para obter mais informações sobre o EventBridge, consulte [O que é o Amazon EventBridge?](#) no Guia do usuário do Amazon EventBridge.

Seguem alguns exemplos de regras do EventBridge para usar com filas do Amazon GameLift:

Faz correspondência com eventos de todas as filas do Amazon GameLift

```
{
  "source": [
    "aws.gamelift"
  ],
  "detail-type": [
    "GameLift Queue Placement Event"
  ]
}
```

Faz correspondência com eventos de uma fila específica

```
{
  "source": [
    "aws.gamelift"
  ],
  "detail-type": [
    "GameLift Queue Placement Event"
  ],
  "resources": [
    "arn:aws:gamelift:your_region:your_account:gamesessionqueue/your_queue_name"
  ]
}
```

Tutorial: Configurar uma fila de sessão de jogo para instâncias spot

Introdução

Este tutorial descreve como configurar o posicionamento de sessões de jogo para jogos implantados em frotas Spot de baixo custo. As frotas spot requerem etapas adicionais para manter a disponibilidade contínua do servidor de jogos para os jogadores.

Público-alvo

Este tutorial é para desenvolvedores de jogos que desejam usar frotas spot para hospedar servidores de jogos personalizados ou servidores em tempo real.

O que você aprenderá

- Defina o grupo de jogadores que a fila de sessões de jogo atende.
- Crie uma infraestrutura de frota para dar suporte ao escopo da fila de sessões de jogo.
- Atribua um alias a cada frota para abstrair o ID da frota.
- Crie uma fila, adicione frotas e priorize onde o Amazon GameLift posiciona as sessões de jogo.
- Adicione políticas de latência do jogador para ajudar a minimizar os problemas de latência.

Pré-requisitos

Antes de criar frotas e filas para o posicionamento das sessões de jogo, execute as seguintes tarefas:

- Analise [Como o Amazon GameLift funciona](#).
- [Integre o servidor de jogos ao Amazon GameLift](#).
- [Carregue a compilação ou o script em tempo real da versão do servidor de jogos](#) no Amazon GameLift.
- [Planeje a configuração da frota](#).

Etapa 1: Definir o escopo da fila

Neste tutorial, criamos uma fila para um jogo que tem uma variação de compilação de servidor de jogos. No lançamento, lançaremos o jogo em dois locais: Ásia-Pacífico (Seul) e Ásia-Pacífico (Singapura). Como esses locais estão próximos um do outro, a latência não é um problema para nossos jogadores.

Neste exemplo, há um segmento de jogadores, o que significa que criamos uma fila. No futuro, quando lançarmos o jogo na América do Norte, poderemos criar uma segunda fila voltada para jogadores norte-americanos.

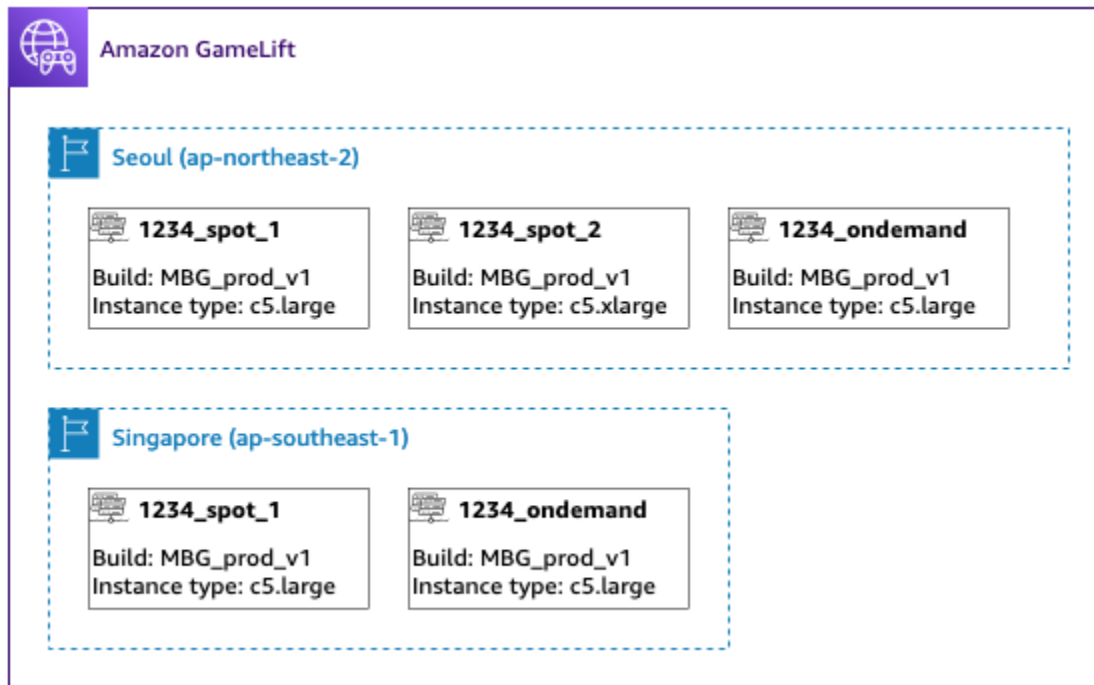
Para obter mais informações, consulte [Defina o escopo da sua fila](#).

Etapa 2: Criar uma infraestrutura de frota spot

Crie frotas em locais e com compilações de servidores de jogos ou scripts que se encaixem no escopo que você definiu em [Etapa 1: Definir o escopo da fila](#).

Neste tutorial, criamos uma infraestrutura de dois locais com, pelo menos, uma frota spot e uma frota sob demanda em cada local. Cada frota implanta a mesma versão de servidor de jogos. Além disso, prevemos que o tráfego de jogadores será mais intenso na localização de Seul e, portanto, adicionamos mais frotas spot lá.

O diagrama a seguir mostra o exemplo da infraestrutura da frota spot, com três frotas na localização ap-northeast-2 (Seul) e duas frotas na localização ap-southeast-1 (Singapura). Todas as instâncias em ambas as frotas estão usando a compilação MBG_Prod_v1. A frota em ap-northeast-2 contém as seguintes configurações de frota: frota 1234_spot_1 com um tipo de instância de c5.large, frota 1234_spot_2 com um tipo de instância de c5.xlarge e frota 1234_ondemand com um tipo de instância de c5.large. A frota em ap-southeast-1 contém as seguintes configurações de frota: frota 1234_spot_1 com um tipo de instância de c5.large e frota 1234_ondemand com um tipo de instância de c5.large.

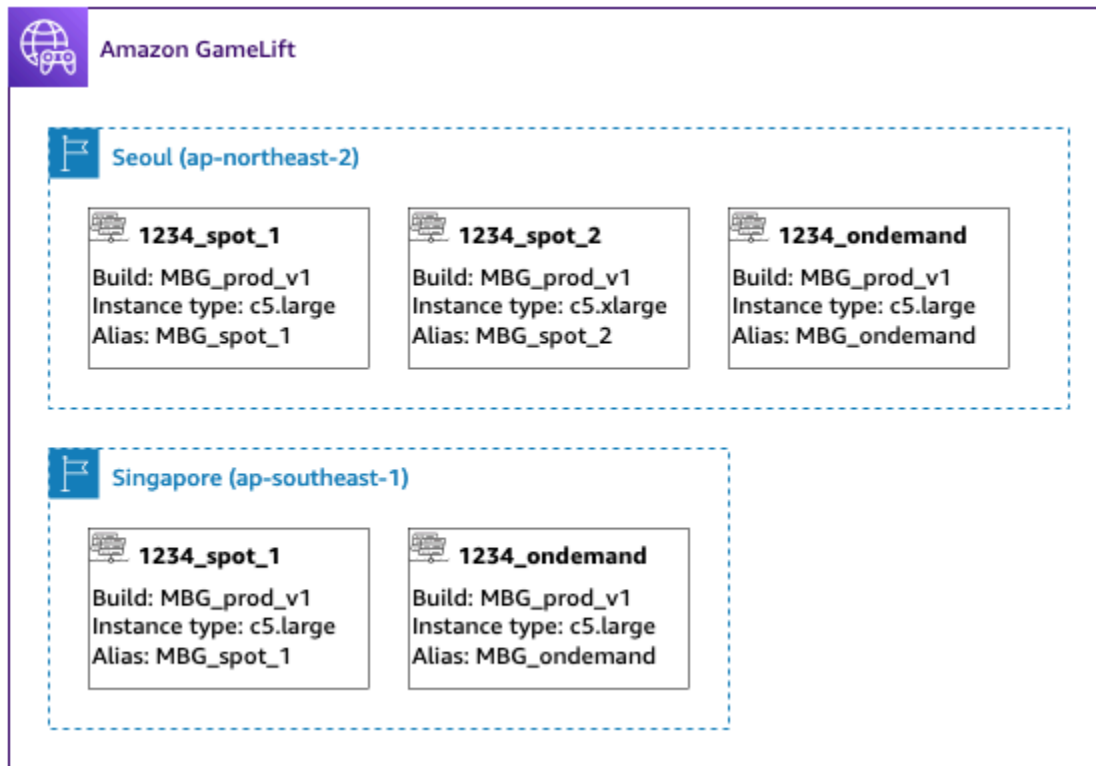


Etapa 3: Atribuir aliases para cada frota

Crie um novo alias para cada frota na sua infraestrutura. Os aliases abstraem as identidades da frota, tornando eficiente a substituição periódica da frota. Para obter mais informações sobre como criar aliases, consulte [Adicionar um alias a uma frota da Amazon GameLift](#).

Nossa infraestrutura de frota tem cinco frotas e, portanto, criamos cinco aliases usando a estratégia de roteamento. Precisamos de três aliases no local Ásia-Pacífico (Seul) e dois aliases no local Ásia-Pacífico (Singapura).

O diagrama a seguir mostra a infraestrutura da frota spot descrita na etapa dois com aliases adicionados a cada frota. A frota **1234_spot_1** tem o alias **MBG_spot_1**, a frota **1234_spot_2** tem o alias **MBG_spot_2** e a frota **1234_ondemand** tem o alias **MBG_ondemand**.



Para obter mais informações, consulte [Criar uma fila de vários locais](#).

Etapa 4: Criar uma fila com destinos

Crie a fila da sessão de jogo e adicione os destinos da frota. Para obter mais informações sobre como criar uma fila, consulte [Criar uma fila de sessões de jogo](#).

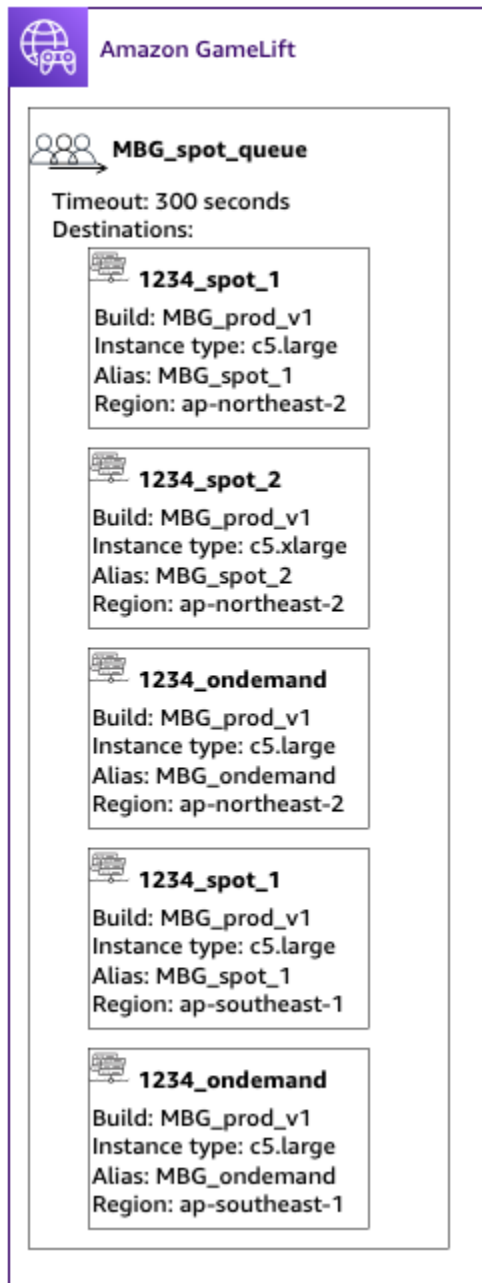
Ao criar sua fila:

- O valor do tempo limite padrão é 10 minutos. Posteriormente, é possível testar como o tempo limite da fila afeta o tempo de espera dos jogadores para entrar nos jogos.
- Por enquanto, pule a seção sobre políticas de latência do jogador. Abordaremos isso na próxima etapa.
- Priorize as frotas na sua fila. Ao trabalhar com frotas spot, recomendamos uma das seguintes abordagens:
 - Se sua infraestrutura usa um local principal com frotas em um segundo local para backup, priorize as frotas primeiro por local e, depois, por tipo de frota.
 - Se sua infraestrutura usa vários locais igualmente, priorize as frotas por tipo de frota, colocando as frotas spot no topo da fila.

Neste tutorial, criamos uma nova fila com o nome **MBG_spot_queue** e adicionamos os aliases das nossas cinco frotas. Em seguida, priorizamos os posicionamentos primeiro por local e, depois, por tipo de frota.

Com base nessa configuração, essa fila sempre tenta posicionar novas sessões de jogo em uma frota spot em Seul. Quando essas frotas estão cheias, a fila usa a capacidade disponível na frota sob demanda de Seul como reserva. Se nenhuma das três frotas de Seul estiver disponível, o Amazon GameLift posicionará as sessões de jogo nas frotas de Singapura.

O diagrama a seguir mostra uma fila com um tempo limite de 300 segundos e destinos priorizados. Os destinos estão na seguinte ordem: 1234_spot_1 em ap-northeast-2, 1234_spot_2 em ap-northeast-2, 1234_ondemand em ap-northeast-2, 1234_spot_1 em ap-southeast-1 e 1234_ondemand em ap-southeast-1.



The screenshot displays the configuration for an Amazon GameLift queue named **MBG_spot_queue**. The queue has a timeout of 300 seconds and is configured with five destinations. Each destination is represented by a card with a small icon and the following details:

- 1234_spot_1**: Build: MBG_prod_v1, Instance type: c5.large, Alias: MBG_spot_1, Region: ap-northeast-2
- 1234_spot_2**: Build: MBG_prod_v1, Instance type: c5.xlarge, Alias: MBG_spot_2, Region: ap-northeast-2
- 1234_ondemand**: Build: MBG_prod_v1, Instance type: c5.large, Alias: MBG_ondemand, Region: ap-northeast-2
- 1234_spot_1**: Build: MBG_prod_v1, Instance type: c5.large, Alias: MBG_spot_1, Region: ap-southeast-1
- 1234_ondemand**: Build: MBG_prod_v1, Instance type: c5.large, Alias: MBG_ondemand, Region: ap-southeast-1

Etapa 5: Adicionar limites de latência à fila

Nosso jogo inclui informações de latência nas solicitações de posicionamento da sessão de jogo. Também temos um atributo de festa de jogadores que cria uma sessão de jogo para um grupo de jogadores. Podemos fazer com que os jogadores esperem um pouco mais para entrar nos jogos com a experiência de jogo ideal. Nossos testes de jogos mostram as seguintes observações:


- Uma latência abaixo de 50 milissegundos é ideal.


- O jogo não pode ser jogado em latências superiores a 250 milissegundos.
- Os jogadores ficam impacientes por volta de um minuto.

Para nossa fila, com um tempo limite de 300 segundos, adicionamos declarações de política que limitam a latência permitida. As declarações de política permitem gradualmente maiores valores de latência de até 250 milissegundos.


Com essa política, nossa fila procura posicionamentos com latência ideal (menos de 50 milissegundos) no primeiro minuto e, depois, relaxa o limite. A fila não faz posicionamentos em que a latência do jogador seja de 250 milissegundos ou mais.


O diagrama a seguir mostra a fila da etapa quatro com as políticas de latência do jogador adicionadas. As políticas de latência do jogador estabelecem o seguinte: imposição do limite de 50 ms por 60 s, imposição do limite de 125 ms por 30 s e imposição do limite de 250 ms até o tempo limite.


**Amazon GameLift**


**MBG_spot_queue**


Timeout: 300 seconds
Destinations:

**1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-northeast-2

**1234_spot_2**
Build: MBG_prod_v1
Instance type: c5.xlarge
Alias: MBG_spot_2
Region: ap-northeast-2

**1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-northeast-2

**1234_spot_1**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_spot_1
Region: ap-southeast-1

**1234_ondemand**
Build: MBG_prod_v1
Instance type: c5.large
Alias: MBG_ondemand
Region: ap-southeast-1

Latency policies:

- Enforce 50ms limit for 60s
- Enforce 125ms limit for 30s
- Enforce 250ms limit until timeout

Resumo

Parabéns! Aqui estão as coisas que você realizou:

- Você tem uma fila de sessões de jogo com escopo definido para um segmento da sua população de jogadores.
- Sua fila usa frotas spot de forma eficaz e é resiliente quando ocorrem interrupções do spot.
- Sua fila prioriza as frotas para uma melhor experiência do jogador.
- A fila tem limites de latência para proteger os jogadores de experiências de jogo ruins.

Agora você pode usar a fila para colocar sessões de jogo para os jogadores que ela atende. Ao fazer solicitações de posicionamento de sessão de jogo para esses jogadores, faça referência ao nome dessa fila de sessões de jogo na solicitação. Para obter mais informações sobre como fazer solicitações de posicionamento de sessão de jogo, consulte [Criar sessões de jogos](#) ou [Integrar um cliente do jogo para servidores em tempo real](#).

Próximas etapas:

- [Projete sua própria fila](#).
- [Crie sua fila](#).
- [Use uma fila com seu cliente de jogo](#).

Gerenciar recursos usando o AWS CloudFormation

É possível usar o AWS CloudFormation para gerenciar os recursos do Amazon GameLift. No AWS CloudFormation, crie um modelo que modele cada recurso e use-o para criar os recursos. Para atualizar recursos, faça alterações no modelo e use o AWS CloudFormation para implementar as atualizações. É possível organizar os recursos em grupos lógicos, chamados pilhas e conjuntos de pilhas.

Usar o AWS CloudFormation para manter os recursos de hospedagem do Amazon GameLift oferece uma maneira mais eficiente de gerenciar conjuntos de recursos da AWS. É possível usar o controle de versão para monitorar alterações de modelo ao longo do tempo e atualizações de coordenadas feitas por vários membros de equipe. Também é possível reutilizar modelos. Por exemplo, ao implantar um jogo em várias regiões, é possível usar o mesmo modelo para criar recursos idênticos em cada região. Também é possível usar esses modelos para implantar o mesmo conjunto de recursos em outra partição.

Para obter mais informações sobre o AWS CloudFormation, consulte o [Guia do usuário do AWS CloudFormation](#). Para visualizar informações de modelo para recursos do Amazon GameLift, consulte a [Referência de tipos de recursos do Amazon GameLift](#).

Práticas recomendadas

Para obter orientações detalhadas sobre uso do AWS CloudFormation, consulte as [melhores práticas do AWS CloudFormation](#) no Guia do usuário do AWS CloudFormation. Além disso, essas melhores práticas têm relevância especial com o Amazon GameLift.

- Gerencie os recursos de forma consistente usando o AWS CloudFormation. Se você alterar os recursos fora do AWS CloudFormation, os recursos ficarão dessincronizados dos modelos de recursos.
- Use pilhas e conjuntos de pilhas do AWS CloudFormation para gerenciar vários recursos com eficiência.
 - Use pilhas para gerenciar grupos de recursos conectados. Por exemplo, uma pilha que contém uma compilação, uma frota que faça referência à compilação e um alias que faça referência à frota. Se você atualizar o modelo para substituir uma compilação, o AWS CloudFormation substituirá as frotas conectadas à compilação. Depois, o AWS CloudFormation atualizará os aliases existentes para apontar para as novas frotas. Para obter mais informações, consulte [Como trabalhar com pilhas](#) no Guia do usuário do AWS CloudFormation.
 - Use conjuntos de pilhas do AWS CloudFormation se estiver implantando pilhas idênticas em várias regiões ou contas da AWS. Para mais informações, consulte [Como trabalhar com conjuntos de pilhas](#) no Guia do usuário do AWS CloudFormation.
- Se estiver usando instâncias spot, inclua uma frota sob demanda como backup. Recomendamos configurar os modelos com duas frotas em cada região, uma frota com instâncias spot e uma frota com instâncias sob demanda.
- Agrupe os recursos específicos do local e os recursos globais em pilhas separadas ao gerenciar recursos em vários locais.
- Posicione os recursos globais próximos aos serviços que os utilizam. Recursos como filas e configurações de criação de jogos tendem a receber um alto volume de solicitações de origens específicas. Ao posicionar os recursos próximos à origem dessas solicitações, você minimiza o tempo de viagem da solicitação e pode melhorar o desempenho geral.
- Coloque a configuração de criação de jogos na mesma região da fila de sessões de jogo usada.
- Crie um alias separado para cada frota na pilha.

Usar pilhas do AWS CloudFormation

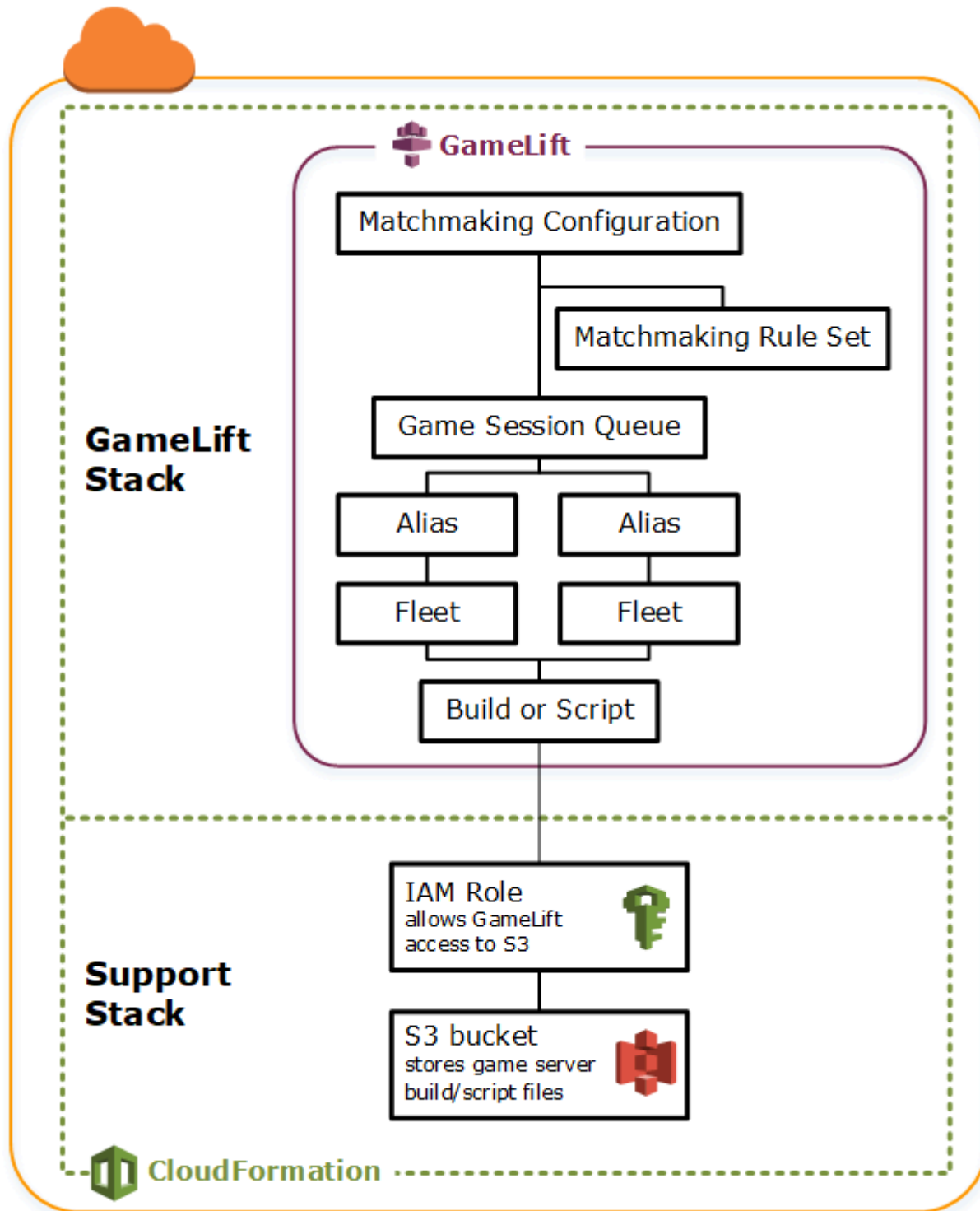
Recomendamos as seguintes estruturas a serem usadas ao configurar pilhas do AWS CloudFormation para recursos do Amazon GameLift. A estrutura de pilha ideal varia dependendo se você está implantando o jogo em um local ou em vários locais.

Pilhas para um único local

Para gerenciar os recursos do Amazon GameLift em um único local, recomendamos uma estrutura de duas pilhas:

- **Pilha de suporte:** essa pilha contém recursos dos quais os recursos do Amazon GameLift dependem. No mínimo, essa pilha deve incluir o bucket do S3 onde você armazena o servidor de jogo personalizado ou arquivos de script Realtime. A pilha também deve incluir um perfil do IAM que conceda ao Amazon GameLift permissão para recuperar os arquivos do bucket do S3 ao criar um recurso de compilação ou de script do Amazon GameLift. Essa pilha também pode conter outros recursos da AWS que são usados com o jogo, como tabelas do DynamoDB, clusters do Amazon Redshift e funções do Lambda.
- **Pilha do Amazon GameLift:** essa pilha contém todos os recursos do Amazon GameLift, incluindo a compilação ou o script, um conjunto de frotas, aliases e a fila de sessão de jogo. O AWS CloudFormation cria um recurso de compilação ou de script com arquivos armazenados no local do bucket do S3 e implanta a compilação ou o script em um ou mais recursos de frota. Cada frota deve ter um alias correspondente. A fila de sessões de jogo faz referência a alguns ou a todos os aliases da frota. Se você estiver usando o FlexMatch para a criação de jogos, essa pilha também conterá uma configuração de criação de jogos e um conjunto de regras.

O diagrama a seguir ilustra uma estrutura de duas pilhas para implantar recursos em uma única região da AWS.



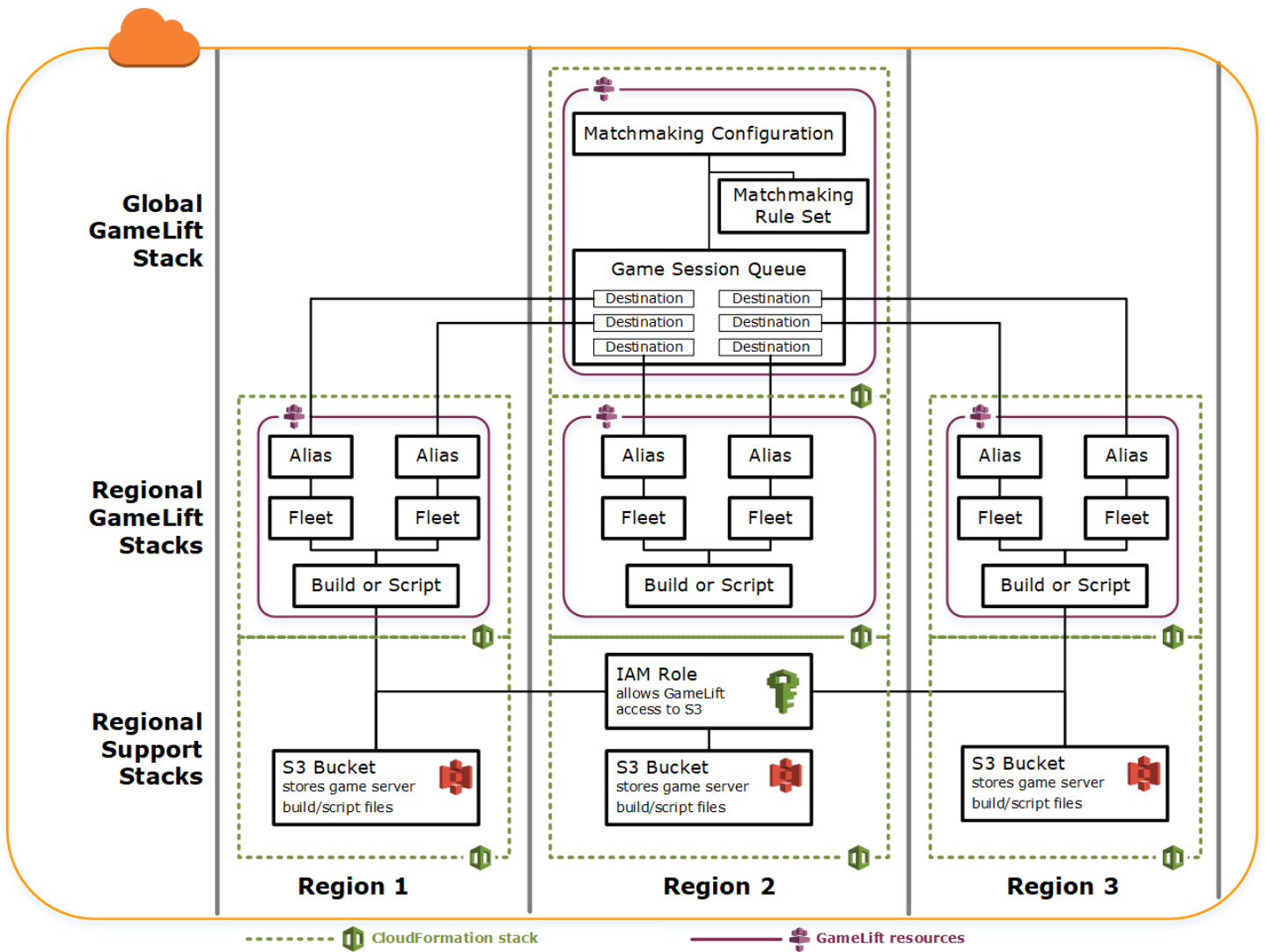
Pilhas para várias regiões

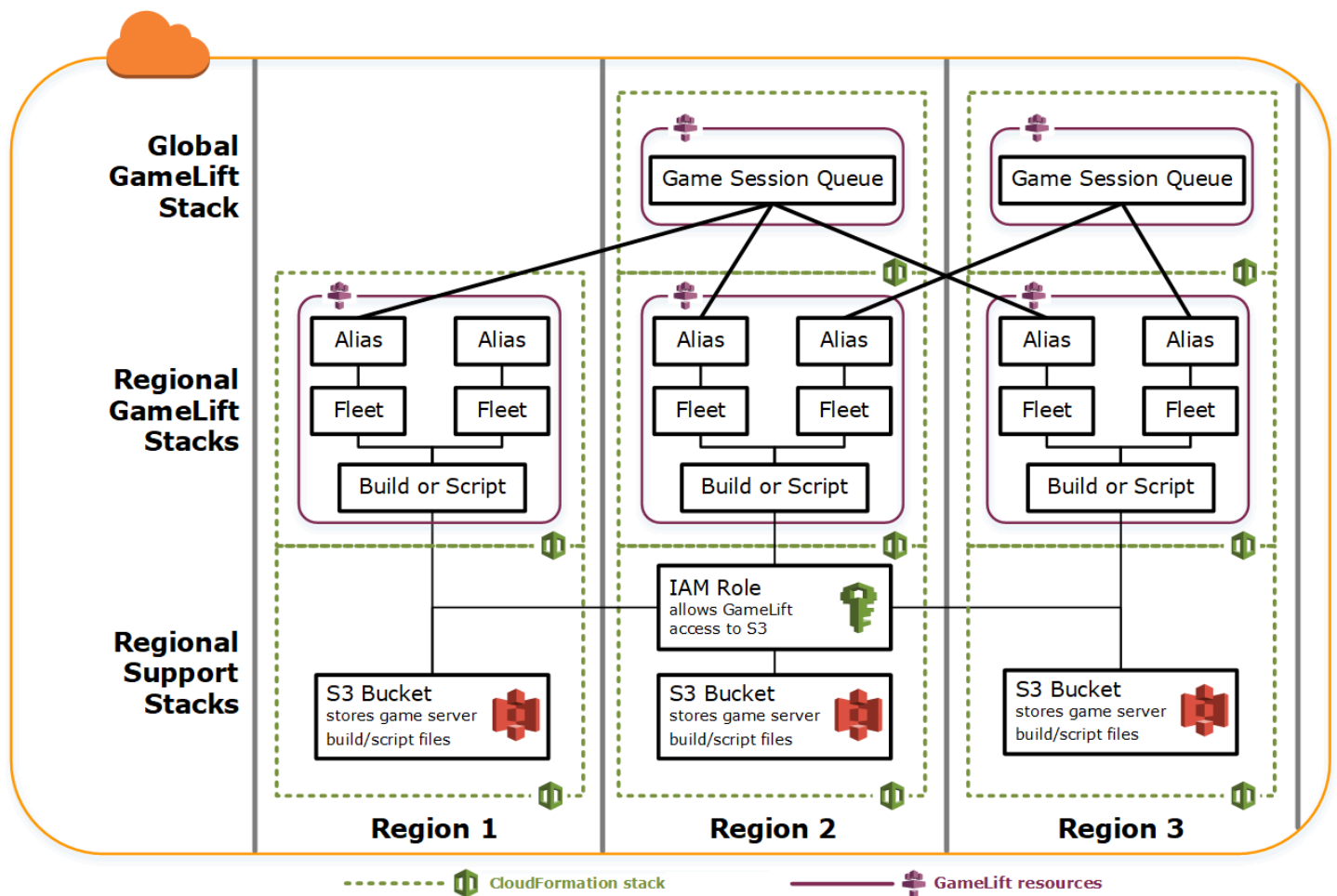
Ao implantar o jogo em mais de uma região, lembre-se de como os recursos podem interagir entre regiões. Alguns recursos, como frotas do Amazon GameLift, só podem fazer referência a outros recursos na mesma região. Outros recursos, como uma fila do Amazon GameLift, são agnósticos em

relação à região. Para gerenciar recursos do Amazon GameLift em várias regiões, recomendamos a estrutura a seguir.

- **Pilhas de suporte regional:** essas pilhas contêm recursos dos quais os recursos do Amazon GameLift depende. Essa pilha deve incluir o bucket do S3 onde você armazena o servidor de jogo personalizado ou os arquivos de script em tempo real. Ela também pode conter outros recursos da AWS para o jogo, como tabelas do DynamoDB, clusters do Amazon Redshift e funções do Lambda. Muitos desses recursos são específicos da região e, sendo assim, você deve criá-los em todas as regiões. O Amazon GameLift também precisa de uma perfil do IAM que permita acesso a esses recursos de suporte. Como um perfil do IAM é agnóstico em relação à região, só é necessário um recurso de função colocado em qualquer região e referenciado em todas as outras pilhas compatíveis.
- **Pilhas regionais do Amazon GameLift:** essa pilha contém os recursos do Amazon GameLift que devem existir em cada região onde o jogo está sendo implantado, incluindo a compilação ou o script, um conjunto de frotas e os aliases. O AWS CloudFormation cria um recurso de compilação ou de script com arquivos em um local de bucket do S3 e implanta a compilação ou o script em um ou mais recursos de frota. Cada frota deve ter um alias correspondente. A fila de sessões de jogo faz referência a alguns ou a todos os aliases da frota. É possível manter um modelo para descrever esse tipo de pilha e usá-lo para criar conjuntos idênticos de recursos em cada região.
- **Pilha global do Amazon GameLift:** essa pilha contém a fila de sessões de jogo e recursos de criação de jogos. Esses recursos podem estar localizados em qualquer região e geralmente são colocados na mesma região. A fila pode fazer referência a frotas ou a aliases que estão localizados em qualquer região. Para colocar filas adicionais em regiões diferentes, crie pilhas globais adicionais.

Os diagramas a seguir ilustram uma estrutura de várias pilhas para implantar recursos em várias regiões da AWS. O primeiro diagrama mostra uma estrutura de uma única fila de sessão de jogo. O segundo diagrama mostra uma estrutura com várias filas.





Atualizar compilações

As compilações do Amazon GameLift são imutáveis, assim como a relação entre uma compilação e uma frota. Como resultado, ao atualizar os recursos de hospedagem para usar um novo conjunto de arquivos de compilação de jogos, é necessário fazer o seguinte:

- Crie uma compilação usando o novo conjunto de arquivos (substituição).
- Crie um conjunto de frotas para implantar a nova compilação do jogo (substituição).
- Redirecione os aliases para apontar para as novas frotas (atualização sem interrupção).

Para obter mais informações, consulte [Atualizar comportamentos de recursos de pilha](#) no Guia do usuário do AWS CloudFormation.

Implantar atualizações de compilação automaticamente

Ao atualizar uma pilha que contém recursos de compilação, frota e alias relacionados, o comportamento padrão do AWS CloudFormation é executar essas etapas automaticamente em sequência. Acione essa atualização primeiro fazendo upload dos novos arquivos de compilação para um novo local do S3. Depois, modifique o modelo de compilação do AWS CloudFormation a fim de apontar para o novo local do S3. Ao atualizar a pilha com o novo local do S3, isso aciona a seguinte sequência do AWS CloudFormation:

1. Recupera os novos arquivos do S3, valida os arquivos e cria uma compilação do Amazon GameLift.
2. Atualiza a referência de compilação no modelo de frota, que aciona a criação da frota.
3. Depois que as novas frotas estiverem ativas, atualiza a referência da frota no alias, o que aciona o alias para atualizar a fim de atingir as novas frotas.
4. Apaga a frota antiga.
5. Exclui a compilação antiga.

Se a fila de sessões de jogo usar aliases de frota, o tráfego de jogadores será automaticamente alterado para as novas frotas assim que os aliases forem atualizados. As frotas antigas são gradualmente drenadas de jogadores à medida que as sessões de jogo terminam. A escalabilidade automática lida com a tarefa de adicionar e remover instâncias de cada conjunto de frotas à medida que o tráfego de jogadores flutua. Como alternativa, é possível especificar uma contagem inicial de instâncias desejada a fim de acelerar rapidamente para a troca e habilitar a escalabilidade automática mais tarde.

Também é possível fazer com que o AWS CloudFormation retenha recursos em vez de excluí-los. Para obter mais informações, consulte [RetainResources](#) na Referência de API do AWS CloudFormation.

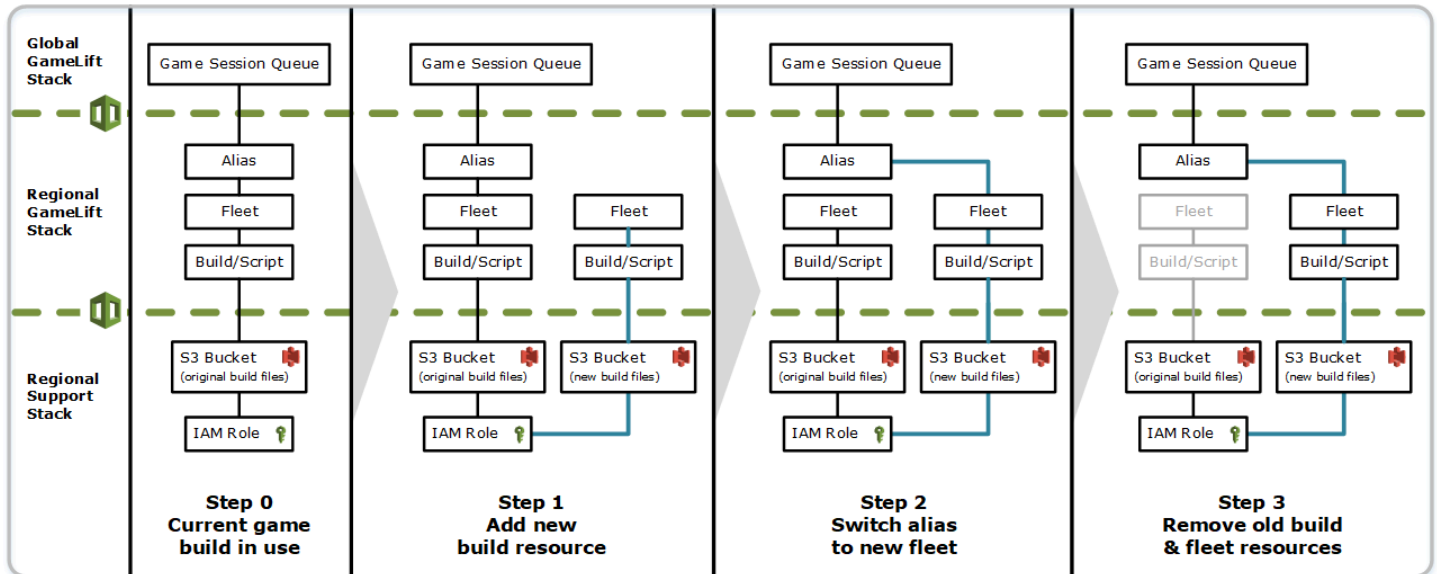
Implantar atualizações de compilação manualmente

Se quiser ter mais controle sobre quando novas frotas são ativadas para os jogadores, há algumas opções. É possível optar por gerenciar aliases manualmente usando o console do Amazon GameLift ou a CLI. Como alternativa, em vez de atualizar o modelo de compilação para substituir as compilações e as frotas, é possível adicionar um segundo conjunto de definições de compilação e de frota ao modelo. Ao atualizar o modelo, o AWS CloudFormation cria um segundo recurso de

compilação e frotas correspondentes. Como os recursos existentes não são substituídos, eles não são excluídos e os aliases permanecem apontando para as frotas originais.

A principal vantagem desta abordagem é que ela oferece flexibilidade. É possível criar recursos separados para a nova versão da compilação, testar os novos recursos e controlar quando a nova frota é acionada para os jogadores. Uma possível desvantagem é que ela requer duas vezes mais recursos em cada região por um breve período.

O diagrama a seguir ilustra esse processo.



Como funcionam as reversões

Ao executar uma atualização de recurso, se qualquer etapa não for concluída com êxito, o AWS CloudFormation iniciará automaticamente uma reversão. Esse processo reverte cada etapa em sequência, excluindo os recursos recém-criados.

Se precisar acionar manualmente uma reversão, altere a chave de localização do S3 do modelo de compilação de volta para o local original e atualize a pilha. Uma compilação e uma frota do Amazon GameLift são criadas, e o alias muda para a nova frota após ela estar ativa. Se estiver gerenciando aliases separadamente, será necessário mudá-los a fim de apontar para as novas frotas.

Para obter mais informações sobre como lidar com uma reversão que falha ou trava, consulte [Continuar revertendo uma atualização](#) no Guia do usuário do AWS CloudFormation.

Emparelhamento de VPC para o Amazon GameLift

Este tópico fornece orientações sobre como configurar uma conexão de emparelhamento da VPC entre seus servidores de jogos hospedados no Amazon GameLift e outros recursos que não sejam do Amazon GameLift. Use a conexão de emparelhamento da nuvem privada virtual (VPC) da Amazon, para permitir que seus servidores de jogos se comuniquem de maneira direta e privada com seus outros recursos da AWS, como um serviço da Web ou um repositório. É possível estabelecer emparelhamento da VPC com todos os recursos que são executados na AWS e gerenciados por uma conta da AWS à qual você tem acesso.

Note

O emparelhamento de VPCs é um recurso avançado. Para saber mais sobre opções de preferência para permitir que os seus servidores de jogos se comuniquem de maneira direta e privada com outros recursos da AWS, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Caso já esteja familiarizado com as VPCs da Amazon e emparelhamento da VPC, compreenda o processo de configuração do emparelhamento com servidores de jogos do Amazon GameLift é um pouco diferente. Você não tem acesso à VPC que contém os servidores de jogos (ela é controlada pelo serviço Amazon GameLift), portanto, não é possível solicitar o emparelhamento da VPC diretamente para ela. Primeiro, pré-autorize a VPC com os recursos que não são do Amazon GameLift para aceitar uma solicitação de emparelhamento do serviço Amazon GameLift. Em seguida, você aciona o Amazon GameLift para solicitar o emparelhamento da VPC, que você acabou de autorizar. O Amazon GameLift realiza as tarefas de criar a conexão de emparelhamento, configurar as tabelas de rotas e configurar a conexão.

Como configurar o emparelhamento de VPC para uma frota existente

1. Obtenha os IDs e credenciais das contas da AWS.

Você precisa de um ID e das credenciais de login para as seguintes contas da AWS. É possível encontrar IDs de contas da AWS fazendo login no [AWS Management Console](#) e visualizando as configurações da sua conta. Para obter as credenciais, acesse o console do IAM.

- A conta da AWS que você usa para gerenciar os servidores de jogos do Amazon GameLift.
- A conta da AWS que você usa para gerenciar os recursos que não são do Amazon GameLift .

Caso esteja usando a mesma conta para recursos do Amazon GameLift e que não são do Amazon GameLift, você precisa do ID e das credenciais dessa conta apenas.

2. Obtenha identificadores para cada VPC.

Obtenha as seguintes informações para as duas VPCs que serão emparelhadas:

- VPC para os servidores de jogos do Amazon GameLift – Esse é o ID da frota do Amazon GameLift. Os servidores de jogos são implantados no Amazon GameLift em uma frota de instâncias do EC2. Cada frota é colocada automaticamente em sua própria VPC, que é gerenciada pelo serviço do Amazon GameLift. Você não tem acesso direto à VPC, e isso é identificado pelo ID da frota.
- VPC para seus recursos da AWS que não são do Amazon GameLift – É possível estabelecer um emparelhamento da VPC com todos os recursos que são executados na AWS e gerenciados por uma conta da AWS à qual você tem acesso. Se você ainda não criou uma VPC para esses recursos, consulte [Conceitos básicos da Amazon VPC](#). Depois de criar uma VPC, você pode encontrar o ID da VPC ao fazer login no [AWS Management Console](#) da Amazon VPC e visualizar suas VPCs.

Note

Ao configurar um emparelhamento, as duas VPCs devem existir na mesma região. A VPC de seu servidor de jogos da frota do Amazon GameLift está na mesma região que a frota.

3. Autorize um emparelhamento de VPC.

Nesta etapa, você pré-autoriza uma solicitação futura do Amazon GameLift para emparelhar a VPC com seus servidores de jogos com sua VPC para recursos que não forem do Amazon GameLift. Esta ação atualiza o grupo de segurança da sua VPC.

Para autorizar o emparelhamento da VPC, chame a API [CreateVpcPeeringAuthorization\(\)](#) de serviço do Amazon GameLift ou use o comando `create-vpc-peering-authorization` da CLI da AWS. Faça esta chamada usando a conta que gerencia seus recursos que não são do Amazon GameLift. Identifique as seguintes informações:

- ID da VPC de emparelhamento – Esse é para a VPC com recursos que não são do Amazon GameLift.
- ID da conta AWS do Amazon GameLift – Essa é a conta que você usa para gerenciar sua frota do Amazon GameLift.

Após autorizar um emparelhamento da VPC, a autorização permanecerá válida por 24 horas, a menos que seja revogada. É possível gerenciar suas autorizações de emparelhamento de VPCs usando as seguintes operações:

- [DescribeVpcPeeringAuthorizations\(\)](#) (CLI `describe-vpc-peering-authorizations` da AWS).
- [DeleteVpcPeeringAuthorization\(\)](#) (CLI `delete-vpc-peering-authorization` da AWS).

4. Solicite uma conexão de emparelhamento.

Com uma autorização válida, é possível solicitar que o Amazon GameLift estabeleça uma conexão de emparelhamento.

Para solicitar um emparelhamento da VPC, chame a API [CreateVpcPeeringConnection\(\)](#) de serviço do Amazon GameLift ou use o comando `create-vpc-peering-connection` da CLI da AWS. Faça essa chamada usando a conta que gerencia seus servidores de jogos do Amazon GameLift. Use as seguintes informações para identificar as duas VPCs que você deseja emparelhar:

- O ID da VPC e o ID da conta da AWS – Essa é a VPC dos recursos que não são do Amazon GameLift e a conta que você usa para gerenciá-los. O ID da VPC deve corresponder ao ID em uma autorização de emparelhamento válida.
- ID da frota – Isto identifica a VPC para os servidores de jogos do Amazon GameLift.

5. Acompanhe o status da conexão de emparelhamento.

Solicitar uma conexão de emparelhamento da VPC é uma operação assíncrona. Para rastrear o status de uma solicitação de emparelhamento e processar os casos de sucesso ou falha, use uma das seguintes opções:

- Sonde continuamente com `DescribeVpcPeeringConnections()`. Essa operação recupera o registro de conexão de emparelhamento da VPC, incluindo o status da solicitação.

Se uma conexão emparelhada for criada com êxito, o registro de conexão também conterá um bloco CIDR de endereços IP privados que é atribuído à VPC.

- Gerencie eventos de frota associados a conexões de emparelhamento das VPCs com [DescribeFleetEvents\(\)](#), incluindo eventos de sucesso e falha.

Uma vez estabelecida a conexão de emparelhamento, é possível gerenciá-la usando as seguintes operações:

- [DescribeVpcPeeringConnections\(\)](#) (CLI `describe-vpc-peering-connections` da AWS).
- [DeleteVpcPeeringConnection\(\)](#) (CLI `delete-vpc-peering-connection` da AWS).

Como configurar o emparelhamento de VPC com uma nova frota

É possível criar uma nova frota do Amazon GameLift e solicitar uma conexão de emparelhamento da VPC ao mesmo tempo.

1. Obtenha os IDs e credenciais das contas da AWS.

Você precisa do ID e das credenciais de login para as seguintes duas contas da AWS. É possível encontrar IDs de contas da AWS fazendo login no [AWS Management Console](#) e visualizando as configurações da sua conta. Para obter as credenciais, acesse o console do IAM.

- A conta da AWS que você usa para gerenciar seus servidores de jogos do Amazon GameLift.
- A conta da AWS que você usa para gerenciar seus recursos que não são do Amazon GameLift.

Caso esteja usando a mesma conta para recursos do Amazon GameLift e de outros recursos que não sejam do Amazon GameLift, precisará do ID e das credenciais dessa conta apenas.

2. Obtenha o ID da VPC para os recursos que não são do Amazon GameLift. AWS

Se você ainda não criou uma VPC para esses recursos, faça isso agora (consulte [Conceitos básicos da Amazon VPC](#)). Certifique-se de criar a nova VPC na mesma região em que planeja criar sua nova frota. Caso os recursos que não sejam do Amazon GameLift forem gerenciados em uma conta da AWS ou um grupo de usuários/usuário diferente daquele com o qual você

usa com o Amazon GameLift,, será necessário usar essas credenciais de conta ao solicitar autorização na próxima etapa.

Depois de criar uma VPC, é possível localizar o ID da VPC no console da Amazon VPC, visualizando suas VPCs.

3. Autorize um emparelhamento da VPC com recursos que não são do Amazon GameLift.

Quando o Amazon GameLift cria a nova frota e a VPC correspondente, ele também envia uma solicitação para emparelhar com a VPC para seus recursos que não são do Amazon GameLift. Você precisa pré-autorizar essa solicitação. Esta etapa atualiza o grupo de segurança da sua VPC.

Ao usar as credenciais da conta que gerenciam os recursos que não são do Amazon GameLift, chame a API [CreateVpcPeeringAuthorization\(\)](#) de serviço do Amazon GameLift ou use o comando `create-vpc-peering-authorization` da CLI da AWS. Identifique as seguintes informações:

- ID da VPC de emparelhamento – O ID da VPC com os recursos que não são do Amazon GameLift.
- ID da conta AWS do Amazon GameLift – ID da conta que você usa para gerenciar sua frota do Amazon GameLift.

Após autorizar um emparelhamento da VPC, a autorização permanecerá válida por 24 horas, a menos que seja revogada. É possível gerenciar suas autorizações de emparelhamento de VPCs usando as seguintes operações:

- [DescribeVpcPeeringAuthorizations\(\)](#) (CLI `describe-vpc-peering-authorizations` da AWS).
- [DeleteVpcPeeringAuthorization\(\)](#) (CLI `delete-vpc-peering-authorization` da AWS).

4. Siga as instruções para [criar uma nova frota usando a CLI da AWS](#). Inclua os parâmetros adicionais a seguir:

- `peer-vpc-aws-account-id` – O ID da conta que você usa para gerenciar a VPC com seus recursos que não são do Amazon GameLift.
- `peer-vpc-id` – ID da VPC com sua conta que não é do GameLift.

Uma chamada bem-sucedida para [create-fleet](#) com os parâmetros de emparelhamento de VPC gera uma nova frota e uma nova solicitação de emparelhamento de VPC. O status da frota é definido para Novo, e o processo de ativação dela é iniciado. O status da solicitação de conexão de emparelhamento é definido como `initiating-request`. Você pode rastrear o sucesso ou a falha da solicitação de emparelhamento chamando [describe-vpc-peering-connections](#).

Ao solicitar uma nova frota e uma conexão de emparelhamento da VPC, ambas as ações serão bem-sucedidas ou falharão. Se uma frota falhar durante o processo de criação, a conexão de emparelhamento da VPC não será estabelecida. Da mesma forma, se uma conexão de emparelhamento da VPC falhar por qualquer motivo, a nova frota falhará ao mudar do status Ativando para Ativo.

Note

A nova conexão de emparelhamento da VPC não será concluída até que a frota esteja pronta para se tornar ativa. Isso significa que a conexão não está disponível e não pode ser usada durante o processo de instalação de compilação do servidor de jogos.

O exemplo a seguir cria uma nova frota e uma conexão de emparelhamento entre uma VPC pré-estabelecida e a VPC para a nova frota. A VPC pré-estabelecida é identificada exclusivamente pela combinação do ID da sua conta da AWS fora do Amazon GameLift e o ID da VPC.

```
$ AWS gamelift create-fleet
  --name "My_Fleet_1"
  --description "The sample test fleet"
  --ec2-instance-type "c5.large"
  --fleet-type "ON_DEMAND"
  --build-id "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff"
  --runtime-configuration "GameSessionActivationTimeoutSeconds=300,
                           MaxConcurrentGameSessionActivations=2,
                           ServerProcesses=[{LaunchPath=C:\game\Bin64.dedicated
\MultiplayerSampleProjectLauncher_Server.exe,
                           Parameters="+sv_port 33435 +start_lobby,
                           ConcurrentExecutions=10}]"
  --new-game-session-protection-policy "FullProtection"
  --resource-creation-limit-policy "NewGameSessionsPerCreator=3,
                                   PolicyPeriodInMinutes=15"
  --ec2-inbound-permissions
  "FromPort=33435, ToPort=33435, IpRange=0.0.0.0/0, Protocol=UDP"
```

```
"FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP"  
  --metric-groups "EMEAfleets"  
  --peer-vpc-aws-account-id "111122223333"  
  --peer-vpc-id "vpc-a11a11a"
```

Versão copiável:

```
AWS gamelift create-fleet --name "My_Fleet_1" --description "The  
sample test fleet" --fleet-type "ON_DEMAND" --metric-groups  
"EMEAfleets" --build-id "build-1111aaaa-22bb-33cc-44dd-5555eeee66ff"  
--ec2-instance-type "c5.large" --runtime-configuration  
"GameSessionActivationTimeoutSeconds=300,MaxConcurrentGameSessionActivations=2,ServerProcesses  
\game\Bin64.dedicated\MultiplayerSampleProjectLauncher_Server.exe,Parameters=  
+sv_port 33435 +start_lobby,ConcurrentExecutions=10}]" --new-game-session-  
protection-policy "FullProtection" --resource-creation-limit-policy  
"NewGameSessionsPerCreator=3,PolicyPeriodInMinutes=15" --ec2-inbound-  
permissions "FromPort=33435,ToPort=33435,IpRange=0.0.0.0/0,Protocol=UDP"  
"FromPort=33235,ToPort=33235,IpRange=0.0.0.0/0,Protocol=UDP" --peer-vpc-aws-account-id  
"111122223333" --peer-vpc-id "vpc-a11a11a"
```

Solucionar problemas de emparelhamento de VPC

Caso esteja com problemas para estabelecer uma conexão de emparelhamento da VPC para os servidores de jogos do Amazon GameLift, considere estas causas raiz comuns:

- Uma autorização para a conexão solicitada não foi encontrada:
 - Verifique o status de uma autorização de VPC para a VPC que não é do Amazon GameLift. Pode não existir ou pode ter expirado.
 - Verifique as regiões das duas VPCs que você está tentando emparelhar. Se não estiverem na mesma região, elas não podem ser emparelhadas.
- Os blocos CIDR (consulte [Configurações inválidas de conexão de emparelhamento da VPC](#)) de suas duas VPCs estão se sobrepondo. Os blocos CIDR de IPv4 atribuídos a VPCs emparelhadas não podem se sobrepor. O bloco CIDR da VPC para sua frota do Amazon GameLift é atribuído automaticamente e não pode ser alterado, portanto, será necessário alterar o bloco CIDR da VPC para seus recursos que não são do Amazon GameLift. Para resolver esse problema:
 - Procure este bloco CIDR para sua frota do Amazon GameLift chamando `DescribeVpcPeeringConnections()`.

- Vá para o console da Amazon VPC, encontre a VPC para seus recursos que não são do Amazon GameLift e altere o bloco CIDR para que eles não se sobreponham.
- A nova frota não foi ativada (ao solicitar o emparelhamento de VPC com uma nova frota). Se a nova frota não conseguir avançar para o status Ativa não haverá VPC com a qual emparelhar, portanto, a conexão de emparelhamento não poderá ser bem-sucedida.

Visualizar os dados do jogo no console

O Amazon GameLift gerenciado coleta continuamente dados de jogos ativos para ajudar a compreender o comportamento e o desempenho do jogador. Com o console do Amazon GameLift, você poderá visualizar, gerenciar e analisar essas informações das compilações, frotas e sessões de jogo e jogador.

Tópicos

- [Veja seu GameLift status atual da Amazon](#)
- [Exibir as compilações](#)
- [Visualizar seus scripts](#)
- [Visualizar as frotas](#)
- [Visualizar detalhes da frota](#)
- [Visualizar dados nas sessões de jogo e jogador](#)
- [Visualizar os aliases](#)
- [Visualizar as filas](#)

Veja seu GameLift status atual da Amazon

O GameLift painel da Amazon fornece uma visão do seguinte:

- O número de compilações nos status Pronto, Inicializado e Falha. Escolha Exibir compilações para obter detalhes sobre as compilações na sua região atual.
- O número de frotas em todos os status. Escolha Exibir frotas para obter detalhes sobre as frotas em sua região atual.
- Seus recursos atuais.
- Anúncios de novos recursos e serviços.

Para abrir o GameLift painel da Amazon

- No [GameLift console da Amazon](#), no painel de navegação, escolha Dashboard.

No painel, você poderá:

- Preparar seu jogo para o lançamento escolhendo Preparar para o lançamento e preencher o questionário de lançamento correspondente.
- Solicitar aumentos de cota de serviço em preparação para lançamentos ou em resposta a lançamentos escolhendo Exibir cotas de serviço.
- Visualizar postagens no blog e informações detalhadas sobre novos recursos escolhendo o link no Destaque de recursos.

☰

GameLift > Dashboard

Dashboard

Build status overview View builds

Viewing data for all builds in N. Virginia region.

✔ Ready
1

⊖ Initialized
0

✘ Failed
0

Fleet status overview View fleets

Viewing data for all fleets in N. Virginia region.

✔ Active
0

⊖ New
0

⊖ Deleting
0

✘ Error
0

⊖ In progress
0

⊖ Terminated
0

Resources (1) View service quotas

Resource type	Count
Builds	1
Scripts	0
Fleets	0
Aliases	0
Queues	0
Matchmaking rule sets	0
Matchmaking configurations	0

Prepare for your game launch [Learn more](#) ↗

Fill out a launch questionnaire

Fill out our game launch questionnaire and email it to the GameLift launch team to ensure a smooth launch. The GameLift launch team will verify your GameLift setup and service limits, preparing you for launch.

[Prepare to launch](#)

Features spotlight

Updates on features available in N. Virginia region

March 22, 2022

Updates to Amazon GameLift FlexMatch for greater flexibility

October 28, 2021

New Asia Pacific (Osaka) region and Graviton2 support for Amazon GameLift

Exibir as compilações

Na página Compilações do [console do Amazon GameLift](#), visualize as informações sobre e gerencie todas as compilações de servidores de jogos enviada para o Amazon GameLift. No painel de navegação, escolha Hospedagem, Compilações.

A página Compilações mostra as seguintes informações para cada compilação:

Note

A página Compilações mostra somente as compilações na sua região atual da AWS.

- Nome – O nome associado à compilação cujo upload foi feito.
- Status – O status da compilação. Exibe uma das três mensagens de status:
 - Inicializado – O upload não foi iniciado ou ainda está em andamento.
 - Pronto – A compilação está pronta para a criação da frota.
 - Falha – O tempo de compilação expirou antes que o Amazon GameLift recebesse os binários.
- Hora de criação – A data e a hora em que você fez o upload da compilação para o Amazon GameLift.
- ID da compilação – ID exclusivo atribuído à compilação no upload.
- Versão – O rótulo da versão associado à compilação cujo upload foi feito.
- Sistema operacional – O sistema operacional no qual a compilação é executada. O sistema operacional de compilação determina qual sistema operacional o Amazon GameLift instala nas instâncias de uma frota.
- Tamanho – o tamanho, em megabytes (MB), do arquivo de compilação cujo upload foi feito para o Amazon GameLift.
- Frotas – O número de frotas implantadas com a compilação.

Nesta página, você poderá fazer o seguinte:

- Visualize detalhes da compilação. Escolha o nome de uma compilação para abrir a página de detalhes da compilação.
- Crie uma nova frota com base em uma compilação. Selecione uma compilação e, em seguida, escolha Criar frota.

- Filtre e classifique a lista de compilações. Use os controles na parte superior da tabela.
- Exclua uma compilação. Selecione uma compilação e escolha Excluir.

Detalhes da compilação

Na página Compilações, escolha o nome de uma compilação para abrir sua página de detalhes. A seção Visão geral da página de detalhes exibe as mesmas informações de resumo da compilação que a página Compilações. A seção Frotas mostra uma lista das frotas criadas com a compilação, incluindo as mesmas informações resumidas da [página Frotas](#).

Visualizar seus scripts

Na página Scripts do [console do Amazon GameLift](#), visualize informações e gerencie todos os scripts de servidores em tempo real que você enviou para o Amazon GameLift. No painel de navegação, escolha Hospedagem, Scripts.

A página Scripts mostra as seguintes informações para cada script:

Note

A página Scripts mostra scripts somente na região atual da AWS.

- Nome – O nome associado ao script carregado.
- ID – O ID exclusivo atribuído ao script no upload.
- Versão – O rótulo da versão associado ao script cujo upload foi feito.
- Tamanho – O tamanho do arquivo de script carregado no Amazon GameLift em megabytes (MB).
- Hora de criação – A data e a hora em que você fez o upload do script para o Amazon GameLift.
- Frotas – O número de frotas implantadas com o script.

Nesta página, você poderá fazer o seguinte:

- Visualize detalhes do script. Escolha o nome de uma compilação para abrir a página de detalhes do script.
- Crie uma nova frota com base em um script. Selecione um script e escolha Criar frota.

- Filtre e classifique a lista de script. Use os controles na parte superior da tabela.
- Exclua um script. Selecione um script e escolha Excluir.

Detalhes do script

Na página Scripts, escolha o nome de um script para abrir sua página de detalhes. A seção Visão geral da página de detalhes exibe as mesmas informações resumidas do script da página Compilações. A seção Frotas mostra uma lista das frotas criadas com o script, incluindo as mesmas informações resumidas da [página Frotas](#).

Visualizar as frotas

Visualize as informações sobre todas as frotas criadas para hospedar os jogos no Amazon GameLift na conta da AWS. A lista mostra as frotas criadas na sua região atual. Na página Fleets (Frotas), crie uma nova frota ou visualize detalhes adicionais sobre uma frota. A [página de detalhes](#) da frota contém informações de uso, métricas, dados da sessão do jogo e dados da sessão do jogador. Também é possível editar um registro de frota ou excluir uma frota.

Para visualizar a página Frotas, escolha Frotas no painel de navegação.

A página Frotas exibe, por padrão, as informações de resumo a seguir. Personalize as informações mostradas escolhendo o botão Configurações (engrenagem).

- Nome – Nome amigável dado para a frota.
- Status – O status atual da frota, que pode ser um destes estados: Nova, Download em andamento, Compilação em andamento e Ativa.
- CHora de criação – A data e hora em que a frota foi criada.
- Tipo de computação – O tipo do computador usado para hospedar os jogos. Uma frota pode ser uma frota EC2 gerenciada ou uma frota Qualquer lugar.
- Instância do tipo – O tipo de instância do Amazon EC2, que determina a capacidade de computação de instâncias da frota.
- Instâncias ativas – O número de instâncias do EC2 em uso para a frota.
- Instâncias desejadas – O número de instâncias do EC2 a serem mantidas ativas.
- Sessões de jogos – O número de sessões de jogos ativas em execução na frota. Os dados têm um atraso de cinco minutos.

Visualizar detalhes da frota

Acesse a página de detalhes Frota pelo painel ou a página Frotas selecionando no nome da frota.

Na página de detalhes da frota, realize as seguintes ações:

- Atualize os atributos, as configurações de porta e a configuração de runtime de uma frota.
- Adicionar ou remover localizações da frota.
- Altere as configurações de capacidade da frota.
- Defina ou altere o ajuste de escala automático do rastreamento de destino.
- Exclua uma frota.

Detalhes

Configurações da frota

- ID da frota – Identificador exclusivo atribuído à frota.
- Nome – O nome da frota.
- ARN – O identificador atribuído à frota. O ARN de uma frota a identifica como um recurso do Amazon GameLift e especifica a região e a conta da AWS.
- Descrição – Uma breve descrição identificável da frota.
- Status – Status atual da frota, que pode ser Nova, Download em andamento, Compilação em andamento e Ativa.
- Hora da criação – A data e a hora em que a frota foi criada.
- Hora de término – A data e a hora em que a frota foi encerrada. Isso ficará em branco se a frota ainda estiver ativa.
- Tipo de frota – Indica se a frota usa instâncias sob demanda ou spot.
- Tipo EC2 – [tipo de instância](#) do Amazon EC2 selecionado para a frota quando ela foi criada.
- Função da instância – Um perfil do IAM da AWS que gerencia o acesso aos seus outros recursos da AWS, caso uma tenha sido fornecida durante a criação da frota.
- Certificado TLS – Se a frota está habilitada ou desativada para usar um certificado TLS para autenticar um servidor de jogos e criptografar toda a comunicação cliente/servidor.
- Grupo de métricas – O grupo usado para agregar métricas para várias frotas.

- Política de proteção de escalabilidade de jogos –Configuração atual para [proteção de sessão de jogo](#) para a frota.
- Máximo de sessões de jogo por jogador – O número máximo de sessões que um jogador pode criar durante o período da Política.
- Período da política – Quanto tempo esperar até redefinir o número de sessões que um jogador criou.

Detalhes da compilação

A seção Detalhes da compilação exibe a compilação hospedada na frota. Selecione o nome da compilação para visualizar a página de detalhes da compilação completa.

Configuração de runtime

A seção Configuração do runtime exibe os processos do servidor a serem executados em cada instância. Ela inclui o caminho do executável do servidor de jogos e os parâmetros de inicialização opcionais.

Ativação da sessão do jogo

A seção Ativação da sessão de jogo exibe o número de processos do servidor que são iniciados ao mesmo tempo e quanto tempo esperar até que o processo seja ativado antes de encerrá-lo.

Configurações de porta do EC2

A seção Portas exibe as permissões de conexão da frota, inclusive os intervalos de configuração do endereço IP e da porta.

Métricas

O guia Métricas exibe uma representação gráfica das métricas da frota ao longo do tempo. Para obter mais informações sobre o uso de métricas no Amazon GameLift, consulte [Monitorar o Amazon GameLift com o Amazon CloudWatch](#).

Eventos

O guia Eventos fornece um log de todos os eventos ocorridos na frota, inclusive o código do evento, a mensagem e o carimbo de data/hora. Consulte Descrições de [Eventos](#) na Referência da API do Amazon GameLift.

Escalabilidade

O guia Escalabilidade contém informações sobre a capacidade da frota, inclusive o status atual e as alterações de capacidade ao longo do tempo. Ela também fornece ferramentas para atualizar os limites de capacidade e gerenciar o ajuste de escala automático.

Escalabilidade da capacidade

Veja as configurações atuais de capacidade da frota para cada localização da frota. Para obter mais informações sobre como alterar limites e capacidade, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#).

- Local da AWS – Nome de um local em que as instâncias da frota são implantadas.
- Status – Status de hospedagem da localização da frota. O status da localização deve ser ACTIVE para poder hospedar jogos.
- Tamanho mínimo – O menor número de instâncias que devem ser implantadas no local.
- Instâncias desejadas – O número alvo de instâncias ativas para manter a localização. Quando as instâncias ativas e as desejadas não são as mesmas, um evento de escalabilidade é iniciado para iniciar ou encerrar as instâncias conforme necessário até que as instâncias ativas sejam iguais às desejadas.
- Tamanho máximo – o maior número de instâncias que podem ser implantadas no local.
- Disponível – O limite de serviço nas instâncias menos o número de instâncias em uso. Esse valor informa o número máximo de instâncias do que você poderá adicionar ao local.

Políticas de ajuste de escala automático

Esta seção aborda as informações sobre as políticas de ajuste de escala automático que são aplicadas à frota. Configure ou atualize uma política com base em destino. As políticas com base em regras da frota, que devem ser definidas usando o SDK da AWS ou a CLI, são exibidas aqui. Para ter mais informações sobre a escalabilidade, consulte [Capacidade de frota em ajuste de escala automático com o Amazon GameLift](#).

Histórico de escalabilidade

Visualize gráficos das mudanças de capacidade ao longo do tempo.

Localizações

O guia Localizações lista todos os locais em que as instâncias da frota são implantadas. Os locais incluem a região de origem da frota e quaisquer locais remotos que tenham sido adicionados. É possível adicionar ou remover locais diretamente nessa guia.

- Local – Nome de um local em que as instâncias da frota são implantadas.
- Status – Status de hospedagem da localização da frota. O status do local rastreia o processo de ativação das primeiras instâncias no local. O status da localização deve ser ACTIVE para poder hospedar jogos.
- Instâncias ativas – O número de instâncias com processos de servidor em execução no local da frota.
- Servidores ativos – O número de processos do servidor de jogos com possibilidade de hospedar sessões de jogo no local da frota.
- Sessões de jogos – O número de sessões de jogos ativas em instâncias no local da frota.
- Sessões de jogadores – O número de sessões de jogadores, que representam jogadores individuais, que estão participando de sessões de jogo ativas na localização da frota.

Sessões de jogos

O guia Sessões de jogos lista as sessões de jogos passadas e atuais hospedadas na frota, incluindo algumas informações detalhadas. Selecione o ID de uma sessão do jogo para acessar informações adicionais da sessão de jogos, inclusive sessões de jogadores. Para obter mais informações sobre sessões de jogadores, consulte [Visualizar dados nas sessões de jogo e jogador](#).

Visualizar dados nas sessões de jogo e jogador

Você poderá visualizar informações sobre as sessões de jogos e jogadores individuais. Para obter mais informações sobre sessões de jogos e sessões de jogadores, consulte [Como os jogadores se conectam aos jogos](#).

Para visualizar a sessão de jogo e os dados do jogador

1. No [console do Amazon GameLift](#), no painel de navegação, escolha Frotas.
2. Escolha a frota na lista de Frotas que hospedou suas sessões de jogo.

3. Escolha a guia Sessões de jogo. Essa guia lista todas as sessões de jogos hospedadas na frota com informações resumidas.
4. Selecione uma sessão de jogo para visualizar informações adicionais sobre a sessão de jogo e uma lista de jogadores conectados ao jogo.

Detalhes

Visão geral

Esta seção exibe um resumo das informações da sua sessão de jogo.

- Status – Status da sessão de jogo.
 - Ativando – A instância está iniciando uma sessão de jogo.
 - Ativo – Uma sessão de jogo está em execução e disponível para receber jogadores, dependendo da [política de criação de jogadores](#) da sessão.
 - Encerrada – A sessão de jogo foi encerrada.
- ARN – O nome do recurso da Amazon da sessão do jogo.
- Nome – Nome gerado para a sessão de jogo.
- Local – O local em que o Amazon GameLift hospedou a sessão do jogo.
- Hora de criação – Data e hora em que o Amazon GameLift criou a sessão de streaming.
- Hora de término – Data e hora em que a sessão do jogo terminou.
- Nome DNS – O nome do host da sessão do jogo.
- Endereço IP – Endereço IP especificado para a sessão de jogo.
- Porta – Número de porta usado para se conectar à sessão de jogo.
- ID do criador – Um identificador exclusivo do jogador que iniciou a sessão do jogo.
- Política de criação de sessão de jogador – Indica se a sessão do jogo está aceitando novos jogadores.
- Política de proteção de escalabilidade de jogos – O tipo de proteção de sessão de jogo a ser definida em todas as novas instâncias que o Amazon GameLift inicia na frota.

Dados de jogo

Dados bem formatados para enviar para sua sessão de jogo no início.

Propriedades de jogo

Propriedades do par de chaves e valores que influenciam sua sessão de jogo.

Dados de marcação de jogos

O JSON de marcador de jogos do FlexMatch. Para revisar e editar o marcador de jogos, escolha Exibir configuração de marcação de jogos. Para obter mais informações sobre a marcação de jogos do FlexMatch, consulte [Compilar um marcador de jogos](#).

Sessões de jogador

Os seguintes dados da sessão de jogador são coletados para cada sessão de jogo:

- ID da sessão do jogador – O identificador atribuído à sessão do jogador.
- ID de jogador – Um identificador exclusivo para o jogador. Selecione nesse ID para obter informações adicionais sobre o jogador.
- Status – O status da sessão de jogador. A seguir estão os status possíveis:
 - Reservado – A sessão de jogador foi reservada, mas os jogadores não estão conectados.
 - Ativo) – A sessão de jogador está conectada atualmente ao servidor de jogos.
 - Concluíd – A sessão de jogador foi encerrada; o jogador não está mais conectado.
 - Tempo limite – O jogador não conseguiu se conectar.
- CHora de início – A hora em que o jogador se conectou à sessão de jogo.
- Hora de término – A hora em que o jogador se desconectou da sessão de jogo.
- Dados do jogador – Informações sobre o jogador fornecidas durante a criação da sessão do jogador.

Informações do jogador

Visualize informações adicionais de um jogador selecionado, inclusive uma lista de todos os jogos a que o jogador se conectou em todas as frotas na região atual. Entre essas informações estão o status, horários de início, horários de término além do tempo total conectado para cada sessão de jogador. Você poderá optar por visualizar dados das sessões de jogos e frotas relevantes.

Visualizar os aliases

A página Alias exibe informações sobre os aliases de frota criada na região atual. Para visualizar a página de aliases, escolha Aliases no painel de navegação.

Na página de aliases, é possível fazer o seguinte:

- Crie um alias. Escolha Criar alias.
- Filtre e classifique a lista de tabelas. Use os controles na parte superior da tabela.
- Visualize detalhes do alias. Selecione um nome de alias para abrir a página de detalhes do alias.
- Exclua um alias. Selecione um alias e, em seguida, selecione Excluir.

Detalhes do alias

A página de detalhes do alias exibe informações sobre o alias.

Nesta página, você poderá:

- Editar um alias. Selecione a opção Editar.
- Visualize as frotas que você associou ao alias.
- Exclua um alias. Selecione a opção Excluir.

Entre as informações detalhadas do alias estão:

- ID – O número exclusivo usado para identificar o alias.
- Descrição – A descrição do alias.
- ARN – O nome do recurso da Amazon do alarme do alias.
- Criação – A data e a hora em que o alias foi criado.
- Última atualização – A data e hora em que o alias foi atualizado pela última vez.
- Tipo de roteamento – O tipo de opção de roteamento para o alias, que pode ser um destes:
 - Simples – Roteia o tráfego do jogador para um ID de frota especificado. Você pode atualizar o ID da frota para um alias a qualquer momento.
 - Terminal – Reenvia uma mensagem para o cliente. Por exemplo, você poderá direcionar jogadores que estejam usando um cliente desatualizado para um local onde eles possam obter uma atualização.
- Tags – pares de chave e valor usados para identificar o alias.

Visualizar as filas

Você pode visualizar informações sobre todas as filas de colocação da sessão do jogo existentes. A página de filas mostra as filas criadas em sua Região atual. Na Queues, você pode criar uma nova fila, excluir filas existentes ou abrir uma página de detalhes de uma fila selecionada. Cada página de detalhes da fila contém os dados de configuração e métricas da fila. Para obter mais informações sobre filas, consulte [Configurar filas do Amazon GameLift para o posicionamento de sessões de jogo](#).

A página de filas exibe as seguintes informações de resumo para cada fila:

- Nome da fila – O nome atribuído à fila. As solicitações de novas sessões de jogos especificam uma fila por esse nome.
- Tempo limite da fila – Tempo máximo, em segundos, que uma solicitação de colocação de sessão de jogo permanece na fila antes do tempo limite.
- Destinos na fila – Número de frotas listadas na configuração da fila. O Amazon GameLift coloca novas sessões de jogo em qualquer frota na fila.

Visualizar detalhes da fila

Você pode acessar informações detalhadas sobre qualquer fila, inclusive a configuração e as métricas da fila. Para abrir uma página de detalhes da fila, acesse a página Filas e escolha um nome de fila.

A página de detalhes da fila exibe uma tabela de resumo e guias contendo informações adicionais. Nesta página, você pode fazer o seguinte:

- Atualize a configuração da fila, a lista de destinos e as políticas de latência de jogador. Selecione a opção Editar.
- Excluir uma fila. Depois de excluir uma fila, todas as solicitações de novas sessões de jogos que referenciem o nome dessa fila falharão. Selecione a opção Excluir.

Note

Para restaurar uma fila excluída, crie uma nova fila com o nome da fila excluída.

Detalhes

Visão geral

A seção Visão geral exibe o nome de recurso da nome do recurso da Amazon (ARN) da fila e o Tempo limite. Você poderá usar o ARN ao referenciar a fila em outras ações ou áreas do Amazon GameLift. O tempo limite é o tempo máximo, em segundos, que uma solicitação de colocação de sessão de jogo permanece na fila antes do tempo limite.

Notificação de eventos

A seção Notificação de eventos lista o tópico do SNS: o Amazon GameLift publica notificações de eventos e os Dados de eventos que são adicionados a todos os eventos criados por essa fila.

Tags

A tabela Tags exibe as chaves e os valores usados para marcar o recurso. Para obter mais informações marcação com tags, consulte [Marcar com tags do recursos da AWS](#).

Métricas

O guia Métricas mostra uma representação gráfica das métricas de fila com o passar do tempo.

As métricas de fila incluem uma variedade de informações que descrevem a atividade de posicionamento na fila, inclusive posicionamentos bem-sucedidos organizados por região. Você poderá usar os dados da região para entender onde você está hospedando seus jogos. As métricas de posicionamento regional podem ajudar a detectar problemas com o design geral da fila.

Métricas de fila também estão disponíveis no Amazon CloudWatch. Para obter descrições das métricas disponíveis, consulte [Métricas do Amazon GameLift para filas](#).

Destinos

O guia Destinos mostra todas as frotas ou aliases listados para a fila.

Quando o Amazon GameLift pesquisa os destinos em busca de recursos disponíveis para hospedar uma nova sessão de jogo, ele pesquisa na ordem listada aqui. Como não há capacidade no primeiro destino listado, o Amazon GameLift coloca novas sessões de jogo lá. As solicitações de colocação de sessões de jogos individuais podem substituir a ordem padrão quando você fornece dados de latência para o jogador. Esses dados dizem ao Amazon GameLift para procurar um destino

disponível com a menor latência média de jogador. Para obter mais informações sobre como projetar suas filas, consulte [Criar uma fila de sessões de jogo](#).

Posicionamento da sessão

Políticas de latência de jogador

A seção Políticas de latência do player mostra todas as políticas que a fila usa. As tabelas listam as políticas na ordem em que são aplicadas.

Locais

A seção Locais mostra os locais em que essa fila pode colocar uma sessão de jogo.

Prioridade

A seção Prioridade mostra a ordem em que a fila avalia os detalhes de uma sessão de jogo.

Ordem de local

A seção Ordem de localização mostra a ordem padrão que a fila usa ao colocar sessões de jogo. A fila usa essa ordem se você não tiver definido outros tipos de prioridade.

Monitorando a Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho da Amazon GameLift e de suas outras AWS soluções. Há três usos principais das métricas com a Amazon GameLift: monitorar a integridade do sistema e configurar alarmes, monitorar o desempenho e o uso do servidor de jogos e gerenciar a capacidade usando o escalonamento manual ou automático.

AWS fornece as seguintes ferramentas de monitoramento para monitorar a Amazon GameLift, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- GameLift Console Amazon
- Amazon CloudWatch — Você pode monitorar GameLift as métricas da Amazon em tempo real, bem como as métricas de outros AWS recursos e aplicativos que você está executando nos AWS serviços. CloudWatch oferece um conjunto de recursos de monitoramento, incluindo ferramentas para criar painéis personalizados e a capacidade de definir alarmes que notificam ou agem quando uma métrica atinge um limite especificado.
- AWS CloudTrail— captura todas as chamadas de API e eventos relacionados feitos por ou em nome de sua AWS conta para a Amazon GameLift e outros AWS serviços. Os dados são fornecidos como arquivos de log para um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram.
- Registros de sessão de jogo – Você poderá enviar mensagens personalizadas do servidor para suas sessões de jogo em arquivos de log armazenados no Amazon S3.

Tópicos

- [Monitorar o Amazon GameLift com o Amazon CloudWatch](#)
- [Log de chamadas de API do Amazon GameLift com o AWS CloudTrail](#)
- [Registro de mensagens do servidor no Amazon GameLift](#)

Monitorar o Amazon GameLift com o Amazon CloudWatch

É possível monitorar o Amazon GameLift usando o Amazon CloudWatch, um serviço da AWS que coleta e processa dados brutos do e os transforma em métricas legíveis quase em tempo real. Essas estatísticas são mantidas por 15 meses para fornecer uma perspectiva histórica sobre o desempenho do seu servidor de jogos com o Amazon GameLift. É possível definir alarmes que observam determinados limites e enviam notificações ou realizam ações quando esses limites são atingidos. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).

A tabela a seguir lista as métricas e as dimensões do Amazon GameLift. Todas as métricas disponíveis no também estão disponíveis no console do Amazon GameLift, que fornece os dados como um conjunto de gráficos personalizável. Para acessar as métricas do CloudWatch para os jogos, use o AWS Management Console, a AWS CLI ou a API do CloudWatch.

Se uma métrica não tiver uma localização, ela usará a localização inicial.

Dimensões para métricas para métricas do Amazon GameLift

O Amazon GameLift dá suporte à filtragem de métricas pelas dimensões a seguir.

Dimensão	Descrição
Location	Filtre métricas para um local de implantação da frota. Se uma métrica não tiver uma localização, ela usará a localização inicial.
FleetId	Filtre métricas para uma única frota. Essa dimensão pode ser usada com todas as métricas de frotas para instâncias, processos de servidor, sessões de jogos e sessões de jogador.
MetricGroup	Filtre métricas para uma coleção de frotas. Adicione uma frota a um grupo de métricas adicionando o nome do grupo de métricas aos atributos da frota (consulte UpdateFleetAttributes()). Essa dimensão pode ser usada com todas as métricas de frotas para instâncias, processos de servidor, sessões de jogos e sessões de jogador.

Dimensão	Descrição
QueueName	Filtre métricas para uma única fila. Essa dimensão é usada apenas com métricas para filas da sessão de jogo.
ConfigurationName	Filtre métricas para uma única configuração de marcação de jogos. Essa dimensão é usada somente com métricas para configurações de marcação de jogos.
ConfigurationName-RuleName	Filtre métricas para uma interseção de uma configuração de marcação de jogos e regra de marcação de jogos. Essa dimensão é usada somente com métricas para regras de marcação de jogos.
InstanceType	Métricas de filtro de uma designação do tipo de instância do EC2, como "c4.large". Essa dimensão é usada somente com métricas para instâncias spot.
OperatingSystem	Filtrar métricas para o sistema operacional de uma instância. Essa dimensão é usada com métricas para instâncias spot.
GameServerGroup	Filtre métricas do FleetIQ para um grupo de servidores de jogos.

Métricas do Amazon GameLift para frotas

O namespace `AWS/GameLift` inclui as métricas a seguir relacionadas a uma atividade em uma frota ou em grupo de frotas. As frotas são usadas com uma solução gerenciada do Amazon GameLift. O serviço do Amazon GameLift envia métricas ao CloudWatch a cada minuto.

Instâncias

Métrica	Descrição
ActiveInstances	<p>Instâncias com status ATIVO, o que significa que estão executando processos de servidores ativos. A contagem inclui instâncias ociosas e aquelas que estão hospedando uma ou mais sessões de jogo. Essa métrica mede a capacidade total da instância atual. Essa métrica pode ser usada com ajuste de escala automático.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
DesiredInstances	<p>Número desejado de instâncias ativas que o Amazon GameLift está trabalhando para manter na frota. Com ajuste de escala automático, esse valor é determinado com base nas políticas de escalabilidade atualmente vigentes. Sem ajuste de escala automático, esse valor é definido manualmente. Essa métrica não está disponível durante a visualização de dados para grupos de métricas da frota.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
IdleInstances	<p>As instâncias ativas que hospedam atualmente zero (0) sessões de jogo. Essa métrica mede a capacidade e disponível, mas não usada. Essa métrica pode ser usada com ajuste de escala automático.</p>

Métrica	Descrição
	<p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
MaxInstances	<p>Número máximo de instâncias permitidas para a frota. O máximo de instâncias de uma frota determina o teto da capacidade durante o aumento manual ou automático. Essa métrica não está disponível durante a visualização de dados para grupos de métricas da frota.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
MinInstances	<p>Número mínimo de instâncias permitidas para a frota. O mínimo de instâncias de uma frota determina o piso da capacidade durante a redução manual ou automática. Essa métrica não está disponível durante a visualização de dados para grupos de métricas da frota.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>

Métrica	Descrição
PercentIdleInstances	<p>Porcentagem de todas as instâncias ativas ociosas (calculadas como <code>IdleInstances / ActiveInstances</code>). Essa métrica pode ser usada para ajuste de escala automático.</p> <p>Unidades: percentual</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
RecycledInstances	<p>Número de instâncias spot que foram recicladas e substituídas. O Amazon GameLift recicla instâncias spot que não estão hospedando sessões de jogo no momento e têm uma alta probabilidade de interrupção.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>
InstanceInterruptions	<p>Número de instâncias spot interrompidas.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>

Métrica	Descrição
CPUUtilization	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. A porcentagem de tempo físico de CPU que o Amazon EC2 usa para executar a instância, que inclui o tempo gasto para executar o código do usuário e o código do Amazon EC2. As ferramentas em seu sistema operacional podem mostrar uma porcentagem em diferente da do CloudWatch devido a fatores como simulação de dispositivos legados, configuração de dispositivos não legados, workloads com interrupções pesadas, migração ao vivo e atualização ao vivo.</p> <p>Unidades: percentual</p>
NetworkIn	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. O número de bytes recebidos em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de rede de entrada para um aplicativo em uma única instância.</p> <p>Unidade: bytes</p>
NetworkOut	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. O número de bytes enviados em todas as interfaces de rede pela instância. Essa métrica identifica o volume de tráfego de rede de saída para um aplicativo em uma única instância.</p> <p>Unidade: bytes</p>

Métrica	Descrição
DiskReadBytes	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. Bytes lidos de todos os volumes de armazenamento de instâncias disponíveis para a instância. Essa métrica é utilizada para determinar o volume de dados que a aplicação lê do disco rígido da instância . Você poderá usá-lo para determinar a velocidade da aplicação.</p> <p>Unidade: bytes</p>
DiskWriteBytes	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. Bytes gravados em todos os volumes de armazenamento de instâncias disponíveis para a instância. Essa métrica é utilizada para determinar o volume de dados que a aplicação grava no disco rígido da instância. Você poderá usá-lo para determinar a velocidade da aplicação.</p> <p>Unidade: bytes</p>
DiskReadOps	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. Operações de leitura concluídas de todos os volumes de armazenamento de instâncias disponíveis para a instância em um período de tempo especificado. Para calcular a média de operações de I/O por segundo (IOPS) para o período, divida o total das operações pelo número de segundos no período em questão.</p> <p>Unidade: contagem</p>

Métrica	Descrição
DiskWriteOps	<p>Métric. do EC2 Para o Amazon GameLift, essa métrica representa o desempenho do hardware em todas as instâncias ativas em um local de frota. Operações de gravação concluídas em todos os volumes de armazenamento de instâncias disponíveis para a instância em um período de tempo especificado. Para calcular a média de operações de I/O por segundo (IOPS) para o período, divida o total das operações pelo número de segundos no período em questão.</p> <p>Unidade: contagem</p>

Processos de servidor

Métrica	Descrição
ActiveServerProcesses	<p>Processos de servidor com status ATIVO, o que significa que estão executando e podem hospedar sessões de jogo. A contagem inclui processos de servidor ociosos e aqueles que estão hospedando sessões de jogo. Essa métrica mede a capacidade total do processo de servidor.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
HealthyServerProcesses	<p>Processos de servidores ativos informados como íntegros. Essa métrica é útil para acompanhar a integridade geral dos servidores de jogo da frota.</p> <p>Unidade: contagem</p>


Métrica	Descrição
<p>PercentHealthyServerProcesses</p>	<p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p> <p>Porcentagem de todos os processos de servidor ativos informados como íntegros (calculada como $\text{HealthyServerProcesses} / \text{ActiveServerProcesses}$).</p> <p>Unidades: percentual</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
<p>ServerProcessAbnormalTerminations</p>	<p>Processos de servidor que foram desligados por causa de circunstâncias anormais desde o relatório mais recente. Essa métrica inclui encerramentos que foram iniciados pelo serviço Amazon GameLift. Isso ocorre quando um processo de servidor para de responder, informa consistente verificações de integridade com falha ou não é encerrado de maneira limpa (chamando ProcessEnding()).</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>

Métrica	Descrição
ServerProcessActivations	<p>Processos de servidor transicionados com êxito do status ACTIVATING para ACTIVE desde o relatório mais recente. Os processos de servidor não podem hospedar sessões de jogo até que estejam ativos.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>
ServerProcessTerminations	<p>Processos de servidor que foram desligados desde o relatório mais recente. Isso inclui todos os processos de servidor transicionados para o status TERMINATED por qualquer motivo, inclusive encerramentos de processo normais e anormais.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>

Sessões de jogos

Métrica	Descrição
ActivatingGameSessions	<p>Sessões de jogo com status ACTIVATING, o que significa que estão no processo de inicialização. As sessões de jogo não podem hospedar jogadores até que estejam ativos. Números altos por um período sustentado podem indicar que as sessões de jogo não estão fazendo a transição do status ACTIVATING</p>

Métrica	Descrição
	<p>G para ACTIVE. Essa métrica pode ser usada com ajuste de escala automático.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
ActiveGameSessions	<p>Sessões de jogo com status ACTIVE, o que significa que eles podem hospedar jogadores e estão hospedando zero ou mais jogadores. Essa métrica mede o número total de sessões de jogo hospedadas no momento. Essa métrica pode ser usada com ajuste de escala automático.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>

Métrica	Descrição
<code>AvailableGameSessions</code>	<p>Processos de servidor ativos e íntegros que não estão sendo usados atualmente para hospedar uma sessão de jogo e podem iniciar uma nova sessão de jogo sem demora para ativar novos processos ou instâncias do servidor. Essa métrica pode ser usada com ajuste de escala automático.</p> <div data-bbox="750 541 1507 999"><p> Note</p><p>Para frotas que limitam as ativações simultâneas de sessões de jogo, use a métrica <code>ConcurrentActiveGameSessions</code>. Essa métrica representa com mais precisão o número de novas sessões de jogo que podem começar sem nenhum tipo de atraso.</p></div> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>

Métrica	Descrição
<p><code>ConcurrentActivatableGameSessions</code></p>	<p>Processos de servidor ativos e saudáveis que não estão sendo usados atualmente para hospedar uma sessão de jogo e podem iniciar imediatamente uma nova sessão de jogo.</p> <p>Essa métrica difere da <code>AvailableGameSessions</code> na seguinte forma: ela não conta os processos do servidor que atualmente não podem ativar uma nova sessão de jogo devido aos limites nas ativações da sessão de jogo. (Consulte a configuração opcional RuntimeConfiguration da frota <code>MaxConcurrentGameSessionActivations</code>). Para frotas que não limitam as ativações de sessões de jogo, essa métrica é idêntica a <code>AvailableGameSessions</code> .</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: localização</p>
<p><code>PercentAvailableGameSessions</code></p>	<p>Porcentagem de slots de sessão de jogo em todos os processos de servidor ativos (íntegros ou não) que não estão sendo usados no momento (calculados como $\frac{\text{AvailableGameSessions}}{[\text{ActiveGameSessions} + \text{AvailableGameSessions} + \text{unhealthy server processes}]}$). Essa métrica pode ser usada com ajuste de escala automático.</p> <p>Unidades: percentual</p> <p>Estatísticas de CloudWatch relevantes: média</p> <p>Dimensões: localização</p>

Métrica	Descrição
GameSessionInterruptions	<p>Número de sessões de jogos em instâncias spot interrompidas.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p> <p>Dimensões: localização</p>

Sessões de jogador

Métrica	Descrição
CurrentPlayerSessions	<p>Sessões de jogador com status ACTIVE (o jogador está conectado a uma sessão de jogo ativa) ou status RESERVED (o jogador recebeu um slot em uma sessão de jogo, mas ainda não se conectou) . Essa métrica pode ser usada com escalabilidade automática.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p>
PlayerSessionActivations	<p>Sessões de jogador transicionados do status RESERVED para ACTIVE desde o relatório mais recente. Isso ocorre quando um jogador se conecta com êxito a uma sessão de jogo ativa.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: soma, média, mínima, máxima</p>

Métricas do Amazon GameLift para filas

O namespace Amazon GameLift inclui as métricas a seguir relacionadas a uma atividade em uma fila de colocação da sessão de jogo. As filas são usadas com uma solução gerenciada do Amazon GameLift. O serviço do Amazon GameLift envia métricas ao CloudWatch a cada minuto.

Métrica	Descrição
AverageWaitTime	<p>Tempo médio em que as solicitações de colocação da sessão de jogo na fila com o status PENDING esperaram ser atendidas.</p> <p>Unidade: segundos</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p> <p>Dimensões: localização</p>
FirstChoiceNotViable	<p>As sessões de jogo que foram posicionadas com êxito, mas NÃO na primeira opção frota, pois esta foi considerada inviável (como uma frota spot com uma alta taxa de interrupção). Essa métrica é baseada no custo, não na latência. A primeira opção de frota se trata da primeira frota listada na fila ou quando uma solicitação de posicionamento inclui dados de latência do jogador, é a primeira frota escolhida pela priorização do FleetIQ. Se não houver frotas spot viáveis, qualquer frota na região podem ser selecionada.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
FirstChoiceOutOfCapacity	<p>As sessões de jogo que foram posicionadas com êxito, mas NÃO na primeira opção frota, pois esta não tinha recursos disponíveis. A primeira opção</p>

Métrica	Descrição
	<p>de frota se trata da primeira frota listada na fila ou quando uma solicitação de posicionamento inclui dados de latência do jogador, é a primeira frota escolhida pela priorização do FleetIQ.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
LowestLatencyPlacement	<p>As sessões de jogo que foram posicionadas com êxito em uma região que oferece a latência mais baixa possível da fila para os jogadores. Essa métrica é emitida apenas quando dados de latência do jogador são incluídos na solicitação de posicionamento.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
LowestPricePlacement	<p>As sessões de jogo que foram posicionadas com êxito em uma frota com o preço mais baixo possível da fila para a região selecionada. Essa frota poderá ser uma frota spot ou uma instância sob demanda se a fila não tiver instâncias spot. Essa métrica é emitida apenas quando dados de latência do jogador são incluídos na solicitação de posicionamento.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>

Métrica	Descrição
<code>Placement <region name></code>	<p>As sessões de jogo que foram posicionadas com êxito em frotas localizadas na região especificada. Essa métrica divide a métrica <code>PlacementSucceeded</code> por região.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
<code>PlacementsCanceled</code>	<p>Solicitações de colocação da sessão de jogo que foram canceladas antes do tempo limite desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
<code>PlacementsFailed</code>	<p>As solicitações de posicionamento da sessão de jogo que por qualquer motivo falharam desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
<code>PlacementsStarted</code>	<p>Novas solicitações de posicionamento da sessão de jogo que foram adicionadas à fila desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>

Métrica	Descrição
PlacementsSucceeded	<p>Solicitações de posicionamento da sessão de jogo que resultaram em uma nova sessão de jogo desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
PlacementsTimedOut	<p>Solicitações de posicionamento da sessão de jogo que alcançaram o tempo limite da fila sem serem atendidas desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
QueueDepth	<p>Número de solicitações de posicionamento de sessão de jogo na fila com status PENDING.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p> <p>Dimensões: localização</p>

Métricas do Amazon GameLift para a marcação de jogos

O namespace `Amazon GameLift` inclui métricas sobre as atividades FlexMatch para as configurações e regras de marcação de jogos. A marcação de jogos do FlexMatch é usada com uma solução gerenciada do Amazon GameLift. O serviço do Amazon GameLift envia métricas ao CloudWatch a cada minuto.

Para obter mais informações sobre a sequência da atividade de marcação de jogos, consulte [Como o Amazon GameLift FlexMatch funciona](#).

Configurações de marcação de jogos

Métrica	Descrição
CurrentTickets	<p>As solicitações de marcação de jogos que estão sendo processadas no momento ou aguardando para serem processadas.</p> <p>Unidade: contagem</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima, soma</p>
MatchAcceptancesTimedOut	<p>Para as configurações de marcação de jogos que exigem aceitação, as correspondências em potencial que expiraram durante a aceitação desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
MatchesAccepted	<p>Para as configurações de marcação de jogos que exigem aceitação, as correspondências em potencial que foram aceitas desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
MatchesCreated	<p>Jogos em potencial que foram criados desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
MatchesPlaced	<p>Jogos que foram colocados com êxito em uma sessão de jogos desde o último relatório.</p> <p>Unidade: contagem</p>

Métrica	Descrição
	Estatísticas relevantes do CloudWatch: soma
MatchesRejected	<p>Para as configurações de marcação de jogos que exigem aceitação, as correspondências em potencial que foram rejeitadas por pelo menos um jogador desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
PlayersStarted	<p>Jogadores em tíquetes de marcação de jogos que foram adicionados desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
TicketsFailed	<p>As solicitações de marcação de jogos que resultaram em uma falha desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
TicketsStarted	<p>As novas solicitações de marcação de jogos que foram criadas desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
TicketsTimedOut	<p>As solicitações de marcação de jogos que atingiram o limite de tempo de espera desde o último relatório.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>

Métrica	Descrição
TimeToMatch	<p>Para solicitações de marcação de jogos que foram colocadas em um jogo em potencial antes do relatório, o tempo entre a criação do tíquete e a criação do jogo em potencial.</p> <p>Unidade: segundos</p> <p>Estatísticas relevantes do CloudWatch: exemplos de dados, média, mínima, máxima</p>
TimeToTicketCancel	<p>Para as solicitações de marcação de jogos que foram canceladas antes do último relatório, o período de tempo entre a criação e o cancelamento do tíquete.</p> <p>Unidade: segundos</p> <p>Estatísticas relevantes do CloudWatch: exemplos de dados, média, mínima, máxima</p>
TimeToTicketSuccess	<p>Para as solicitações de marcação de jogos que foram bem-sucedidas antes do último relatório, o período de tempo entre a criação e a disposição de jogos bem-sucedidas.</p> <p>Unidade: segundos</p> <p>Estatísticas relevantes do CloudWatch: exemplos de dados, média, mínima, máxima</p>

Regras de marcação de jogos

Métrica	Descrição
RuleEvaluationsPassed	<p>Avaliações da regra durante o processo de marcação de jogos que passaram desde o último</p>

Métrica	Descrição
	<p>relatório. Essa métrica é limitada às 50 regras principais.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>
RuleEvaluationsFailed	<p>Avaliações da regra durante a marcação de jogos que falharam desde o último relatório. Essa métrica é limitada às 50 regras principais.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p>

Métricas do Amazon GameLift para FleetIQ

O namespace `Amazon GameLift` inclui métricas para o grupo de servidores de jogos do FleetIQ e a atividade do servidor de jogos como parte de uma solução autônoma do FleetIQ para hospedagem de jogos. O serviço do Amazon GameLift envia métricas ao CloudWatch a cada minuto. Consulte também o [Monitorar grupos e instâncias do Auto Scaling usando o Amazon CloudWatch](#) no Guia do usuário do Amazon EC2 Auto Scaling.

Métrica	Descrição
AvailableGameServers	<p>Servidores de jogos que estão disponíveis para iniciar uma execução de jogo e não estão ocupados atualmente com uma partida. Esse número inclui servidores de jogos que foram reivindicados, mas ainda estão com status AVAILABLE.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup</p>

Métrica	Descrição
<code>UtilizedGameServers</code>	<p>Servidores de jogos que estão ocupados com uma partida no momento. Esse número inclui servidores de jogos que estão no status UTILIZED.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup</p>
<code>DrainingAvailableGameServers</code>	<p>Servidores de jogos em instâncias agendadas para encerramento que não oferecem suporte à jogabilidade no momento. Esses servidores de jogos são a prioridade mais baixa a ser reivindicada em resposta a uma nova solicitação de reivindicação.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup</p>
<code>DrainingUtilizedGameServers</code>	<p>Servidores de jogos em instâncias agendadas para encerramento que estão oferecendo suporte à jogabilidade no momento.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup</p>

Métrica	Descrição
PercentUtilizedGameServers	<p>Parte dos servidores de jogos que estão oferecendo o suporte a execuções de jogos no momento. Essa métrica indica a quantidade de capacidade do servidor de jogos que está em uso no momento. Ela é útil para gerar uma política de ajuste de escala automático que pode adicionar e remover instâncias dinamicamente para que correspondam à demanda de jogadores.</p> <p>Unidades: percentual</p> <p>Estatísticas de CloudWatch relevantes: média, mínima, máxima</p> <p>Dimensões: GameServerGroup</p>
GameServerInterruptions	<p>Servidores de jogos em instâncias spot que foram interrompidos devido à disponibilidade limitada do spot.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup, InstanceType</p>
InstanceInterruptions	<p>Instâncias spot que foram interrompidas devido à disponibilidade limitada.</p> <p>Unidade: contagem</p> <p>Estatísticas relevantes do CloudWatch: soma</p> <p>Dimensões: GameServerGroup, InstanceType</p>

Log de chamadas de API do Amazon GameLift com o AWS CloudTrail

O Amazon GameLift é integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, por função ou por um serviço da AWS no Amazon GameLift. O CloudTrail captura todas as chamadas de API para o Amazon GameLift como eventos. As chamadas capturadas incluem as chamadas do console do Amazon GameLift e as chamadas de código para as operações da API do Amazon GameLift. Caso crie uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon GameLift. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Histórico de eventos. Usar as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita ao Amazon GameLift, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e outros detalhes.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

Informações sobre o Amazon GameLift no CloudTrail

O CloudTrail é habilitado em sua Conta da AWS quando ela é criada. Quando ocorre uma atividade no Amazon GameLift, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviço da AWS no Histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua Conta da AWS. Para obter mais informações, consulte [Como visualizar eventos com o histórico de eventos do CloudTrail](#).

Para obter um registro de eventos em andamento na sua Conta da AWS, incluindo eventos do Amazon GameLift, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Regiões da AWS. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços AWS para analisar mais ainda e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Receber arquivos de log do CloudTrail de várias regiões](#) e [Receber arquivos de log do CloudTrail de várias contas](#)

Todas as ações do Amazon GameLift são registradas pelo CloudTrail e documentadas na [Referência da API do Amazon GameLift](#). Por exemplo, as chamadas para as ações `CreateGameSession`, `CreatePlayerSession` e `UpdateGameSession` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte [Elemento de identidade do usuário do CloudTrail](#).

Noções básicas sobre entradas do arquivo de log do Amazon GameLift

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra as ações `CreateFleet` e `DescribeFleetAttributes`.

```
{
  "Records": [
    {
      "eventVersion": "1.04",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/myUserName",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "myUserName"
      },
    },
  ],
}
```

```
"eventTime": "2015-12-29T23:40:15Z",
"eventSource": "gamelift.amazonaws.com",
"eventName": "CreateFleet",
"awsRegion": "us-west-2",
"sourceIPAddress": "192.0.2.0",
"userAgent": "[]",
"requestParameters": {
  "buildId": "build-92b6e8af-37a2-4c10-93bd-4698ea23de8d",
  "e2InboundPermissions": [
    {
      "ipRange": "10.24.34.0/23",
      "fromPort": 1935,
      "protocol": "TCP",
      "toPort": 1935
    }
  ],
  "logPaths": [
    "C:\\game\\serverErr.log",
    "C:\\game\\serverOut.log"
  ],
  "e2InstanceType": "c5.large",
  "serverLaunchPath": "C:\\game\\MyServer.exe",
  "description": "Test fleet",
  "serverLaunchParameters": "-paramX=baz",
  "name": "My_Test_Server_Fleet"
},
"responseElements": {
  "fleetAttributes": {
    "fleetId": "fleet-0bb84136-4f69-4bb2-bfec-a9b9a7c3d52e",
    "serverLaunchPath": "C:\\game\\MyServer.exe",
    "status": "NEW",
    "logPaths": [
      "C:\\game\\serverErr.log",
      "C:\\game\\serverOut.log"
    ],
    "description": "Test fleet",
    "serverLaunchParameters": "-paramX=baz",
    "creationTime": "Dec 29, 2015 11:40:14 PM",
    "name": "My_Test_Server_Fleet",
    "buildId": "build-92b6e8af-37a2-4c10-93bd-4698ea23de8d"
  }
},
"requestID": "824a2a4b-ae85-11e5-a8d6-61d5cafb25f2",
"eventID": "c8fbea01-fbf9-4c4e-a0fe-ad7dc205ce11",
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.04",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "AIDACKCEVSQ6C2EXAMPLE",
      "arn": "arn:aws:iam::111122223333:user/myUserName",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "userName": "myUserName"
    },
    "eventTime": "2015-12-29T23:40:15Z",
    "eventSource": "gamelift.amazonaws.com",
    "eventName": "DescribeFleetAttributes",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "[]",
    "requestParameters": {
      "fleetIds": [
        "fleet-0bb84136-4f69-4bb2-bfec-a9b9a7c3d52e"
      ]
    },
    "responseElements": null,
    "requestID": "82e7f0ec-ae85-11e5-a8d6-61d5cafb25f2",
    "eventID": "11daabcb-0094-49f2-8b3d-3a63c8bad86f",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
]
}
```

Registro de mensagens do servidor no Amazon GameLift

Você poderá capturar mensagens personalizadas do servidor dos seus servidores do Amazon GameLift em arquivos de log. A forma como você configura o registro depende do uso de servidores personalizados ou servidores em tempo real (consulte as subseções apropriadas neste capítulo).

Tópicos

- [Registro em log de mensagens do servidor \(servidores personalizados\)](#)

- [Registrando mensagens do servidor \(servidores em tempo real\)](#)

Registro em log de mensagens do servidor (servidores personalizados)

Você pode capturar mensagens personalizadas do servidor de seus servidores GameLift personalizados da Amazon em arquivos de log. Para saber mais sobre o registro em servidores em tempo real, consulte [Registrando mensagens do servidor \(servidores em tempo real\)](#).

Important

Há um limite no tamanho de um arquivo de log por sessão de jogo (veja [GameLift endpoints e cotas da Amazon](#) no Referência geral da AWS). Quando uma sessão de jogo termina, a Amazon GameLift carrega os registros do servidor para o Amazon Simple Storage Service (Amazon S3). A Amazon não GameLift fará upload de registros que excedam o limite. Os troncos podem crescer muito rapidamente e exceder o limite de tamanho. Você deve monitorar seus logs e limitar a saída do log somente às mensagens necessárias.

Configurando o registro em log para servidores personalizados

Com os servidores GameLift personalizados da Amazon, você escreve seu próprio código para realizar o registro, que você configura como parte da configuração do processo do servidor. A Amazon GameLift usa sua configuração de registro para identificar os arquivos que devem ser enviados para o Amazon S3 no final de cada sessão de jogo.

As instruções a seguir mostram como configurar o registro usando exemplos de código simplificados:

C++

Para configurar o registro em log (C++)

1. Crie um vetor de strings que sejam caminhos de diretório para arquivos de log do servidor de jogos.

```
std::string serverLog("serverOut.log");           // Example server log file
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
```

2. Forneça seu vetor como o [LogParameters](#) do seu [ProcessParameters](#) objeto.

```
Aws::GameLift::Server::ProcessParameters processReadyParameter =
  Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::onHealthCheck, this),
    std::bind(&Server::onUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths));
```

3. Forneça o [ProcessParameters](#) objeto ao chamar [ProcessReady\(\)](#).

```
Aws::GameLift::GenericOutcome outcome =
  Aws::GameLift::Server::ProcessReady(processReadyParameter);
```

Para um exemplo mais completo, consulte [ProcessReady\(\)](#).

C#

Para configurar o registro em log (C#)

1. Crie uma lista de strings que são caminhos de diretório para arquivos de log do servidor de jogos.

```
List<string> logPaths = new List<string>();
logPaths.Add("C:\\game\\serverOut.txt");    // Example of a log file that the
game server writes
```

2. Forneça sua lista como a [LogParameters](#) do seu [ProcessParameters](#) objeto.

```
var processReadyParameter = new ProcessParameters(
  this.OnGameSession,
  this.OnProcessTerminate,
  this.OnHealthCheck,
  this.OnGameSessionUpdate,
  port,
  new LogParameters(logPaths));
```

3. Forneça o [ProcessParameters](#) objeto ao chamar [ProcessReady\(\)](#).

```
var processReadyOutcome =
```

```
GameLiftServerAPI.ProcessReady(processReadyParameter);
```

Para um exemplo mais completo, consulte [ProcessReady\(\)](#).

Gravar em registros

Seus arquivos de log existem após o início do processo do servidor. Você poderá gravar nos registros usando qualquer método para gravar nos arquivos. Para capturar toda a saída padrão e a saída de erro do seu servidor, remapeie os fluxos de saída para arquivos de log, como nos exemplos a seguir:

C++

```
std::freopen("serverOut.log", "w+", stdout);  
std::freopen("serverErr.log", "w+", stderr);
```

C#

```
Console.SetOut(new StreamWriter("serverOut.txt"));  
Console.SetError(new StreamWriter("serverErr.txt"));
```

Acessando logs do servidor

Quando uma sessão de jogo termina, a Amazon armazena GameLift automaticamente os registros em um bucket do Amazon S3 e os retém por 14 dias. Para obter a localização dos registros de uma sessão de jogo, você pode usar a operação da [GetGameSessionLogUrl](#) API. Para baixar os registros, use a URL que a operação retorna.

Registrando mensagens do servidor (servidores em tempo real)

Você pode capturar mensagens personalizadas do servidor de seus servidores em tempo real em arquivos de log. Para saber mais sobre o registro em servidores personalizados, consulte [Registro em log de mensagens do servidor \(servidores personalizados\)](#).

Há diferentes tipos de mensagens que você pode enviar para seus arquivos de log (consulte [Registrar mensagens no script do seu servidor](#)). Além das mensagens personalizadas, seus servidores em tempo real emitem mensagens do sistema usando os mesmos tipos de mensagem e

gravam nos mesmos arquivos de log. Você pode ajustar o nível de registro da sua frota para reduzir a quantidade de mensagens de registro que seus servidores geram (consulte [Ajustar logs](#)).

⚠ Important

Há um limite no tamanho de um arquivo de log por sessão de jogo (veja [GameLift endpoints e cotas da Amazon](#) no Referência geral da AWS). Quando uma sessão de jogo termina, a Amazon GameLift carrega os registros do servidor para o Amazon Simple Storage Service (Amazon S3). A Amazon não GameLift fará upload de registros que excedam o limite. Os troncos podem crescer muito rapidamente e exceder o limite de tamanho. Você deverá monitorar seus registros e limitar a saída do log somente às mensagens necessárias.

Registrar mensagens no script do seu servidor

Você pode gerar mensagens personalizadas no [script para seus servidores em tempo real](#). Use as etapas a seguir para enviar mensagens do servidor para um arquivo de log:

1. Crie uma variável para manter a referência ao objeto logger.

```
var logger;
```

2. Na função de `init()`, obtenha o registrador do objeto de sessão e atribua-o à sua variável do registrador.

```
function init(rtSession) {  
    session = rtSession;  
    logger = session.getLogger();  
}
```

3. Chame a função apropriada no registrador para gerar uma mensagem.

Mensagens de depuração

```
logger.debug("This is my debug message...");
```

Mensagens informativas

```
logger.info("This is my info message...");
```


Mensagens de aviso

```
logger.warn("This is my warn message...");
```

Mensagens de erro

```
logger.error("This is my error message...");
```

Mensagens de erro fatais

```
logger.fatal("This is my fatal error message...");
```

Mensagens de erro fatais da experiência do cliente

```
logger.cxfatal("This is my customer experience fatal error message...");
```

Para obter um exemplo das instruções de registro em um script, consulte [Exemplo de script de servidores em tempo real](#).

A saída nos arquivos de log indica o tipo de mensagem (DEBUG,INFO,WARN,ERROR,FATAL,CXFATAL), conforme mostrado nas seguintes linhas de um exemplo de registro:

```
09 Sep 2021 11:46:32,970 [INFO] (gamelift.js) 215: Calling GameLiftServerAPI.InitSDK...
09 Sep 2021 11:46:32,993 [INFO] (gamelift.js) 220: GameLiftServerAPI.InitSDK succeeded
09 Sep 2021 11:46:32,993 [INFO] (gamelift.js) 223: Waiting for Realtime server to
start...
09 Sep 2021 11:46:33,15 [WARN] (index.js) 204: Connection is INSECURE. Messages will be
sent/received as plaintext.
```

Acessando logs do servidor

Quando uma sessão de jogo termina, a Amazon armazena GameLift automaticamente os registros no Amazon S3 e os retém por 14 dias. Você pode usar a [chamada de GetGameSessionLogUrl API](#) para obter a localização dos registros de uma sessão de jogo. Use o URL retornado pela chamada de API para baixar os registros.

Ajustar logs

Os troncos podem crescer muito rapidamente e exceder o limite de tamanho. Você deverá monitorar seus registros e limitar a saída do log somente às mensagens necessárias. Para servidores em tempo real, você poderá ajustar o nível de registro fornecendo um parâmetro na configuração de runtime da sua frota no formulário `loggingLevel:LOGGING_LEVEL`, onde `LOGGING_LEVEL` está um dos seguintes valores:

1. `debug`
2. `info` (padrão)
3. `warn`
4. `error`
5. `fatal`
6. `cxfatal`

Essa lista está ordenada da menos severa (`debug`) à mais grave (`cxfatal`). Você define um único `loggingLevel` e o servidor só registrará mensagens nesse nível de gravidade ou em um nível de gravidade mais alto. Por exemplo, a configuração `loggingLevel:error` fará com que todos os servidores da sua frota apenas gravem `error`, `fatal`, e `cxfatal` mensagens ao log.

É possível definir o nível de registro da frota ao criá-lo ou depois que ele já existe. Alterar o nível de registro da sua frota após sua execução afetará apenas os registros das sessões de jogo criadas após a atualização. Os registros de qualquer sessão de jogo existente não serão afetados. Se você não definir um nível de registro ao criar sua frota, seus servidores definirão o nível de registro como padrão `info`. Consulte as seções a seguir para obter instruções sobre como definir o nível de registro.

Definindo o nível de registro ao criar uma frota de servidores em tempo real (console)

Siga as instruções em [Criar uma frota gerenciada pelo Amazon GameLift](#) para criar sua frota, com a seguinte adição:

- Na subetapa de Alocação de processos do servidor da etapa de Gerenciamento de processos, forneça o par de valores-chave do nível de registro (como `loggingLevel:error`) como um valor para os Parâmetros de inicialização. Use um caractere não alfanumérico (exceto vírgula) para separar o nível de registro de quaisquer parâmetros adicionais (por exemplo, `loggingLevel:error +map Winter444`).

Definindo o nível de registro ao criar uma frota de servidores em tempo real (AWS CLI)

Siga as instruções em [Criar uma frota gerenciada pelo Amazon GameLift](#) para criar sua frota, com a seguinte adição:

- No argumento do parâmetro `--runtime-configuration` para [create-fleet](#), forneça o par de valores-chave do nível de registro (como `loggingLevel:error`) como um valor para `Parameters`. Use um caractere não alfanumérico (exceto vírgula) para separar o nível de registro de quaisquer parâmetros adicionais. Veja o exemplo a seguir:

```
--runtime-configuration "GameSessionActivationTimeoutSeconds=60,  
                          MaxConcurrentGameSessionActivations=2,  
                          ServerProcesses=[{LaunchPath=/local/game/myRealtimeLaunchScript.js,  
                                              Parameters=loggingLevel:error +map Winter444,  
                                              ConcurrentExecutions=10}]"
```

Definindo o nível de registro para uma frota de servidores em tempo real em execução (console)

Siga as instruções em [Atualizar uma configuração de frota](#) para atualizar sua frota usando o Amazon GameLift Console, com a seguinte adição:

- Na página Editar frota, em Alocação de processos do servidor, forneça o par de valores-chave do nível de registro (como `loggingLevel:error`) como um valor para os Parâmetros de inicialização. Use um caractere não alfanumérico (exceto vírgula) para separar o nível de registro de quaisquer parâmetros adicionais (por exemplo, `loggingLevel:error +map Winter444`).

Definindo o nível de registro para uma frota de servidores em tempo real em execução (AWS CLI)

Siga as instruções em [Atualizar uma configuração de frota](#) para atualizar sua frota usando o AWS CLI, com a seguinte adição:

- No argumento do parâmetro `--runtime-configuration` para [update-runtime-configuration](#), forneça o par de valores-chave do nível de registro (como `loggingLevel:error`) como um valor para `Parameters`. Use um caractere não alfanumérico (exceto vírgula) para separar o nível de registro de quaisquer parâmetros adicionais. Veja o exemplo a seguir:

```
--runtime-configuration "GameSessionActivationTimeoutSeconds=60,
```

```
MaxConcurrentGameSessionActivations=2,  
ServerProcesses=[{LaunchPath=/local/game/myRealtimeLaunchScript.js,  
Parameters=loggingLevel:error +map Winter444,  
ConcurrentExecutions=10}]"
```

Segurança na Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

A segurança na nuvem AWS é a maior prioridade. Como cliente da AWS, você se beneficiará de data centers e arquiteturas de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a segurança da nuvem e segurança na nuvem:

- Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos [AWS programas](#) de de . Para saber mais sobre os programas de conformidade que se aplicam à Amazon GameLift, consulte [AWS serviços no escopo por programa de conformidade AWS](#) .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e os AWS regulamentos aplicáveis.

Essa documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a Amazon GameLift. Os tópicos a seguir mostram como configurar a Amazon GameLift para atender aos seus objetivos de segurança e conformidade. Você também aprende a usar outros AWS serviços que ajudam você a monitorar e proteger seus GameLift recursos da Amazon.

Tópicos

- [Proteção de dados na Amazon GameLift](#)
- [Gerenciamento de identidade e acesso para a Amazon GameLift](#)
- [Registro em log e monitoramento com o Amazon GameLift](#)
- [Validação de conformidade para a Amazon GameLift](#)
- [Resiliência na Amazon GameLift](#)
- [Segurança da infraestrutura na Amazon GameLift](#)
- [Análise de configuração e vulnerabilidade na Amazon GameLift](#)

- [Melhores práticas de segurança para a Amazon GameLift](#)

Proteção de dados na Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

O [modelo de responsabilidade AWS compartilhada](#) de se aplica à proteção de dados na Amazon GameLift. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para ter mais informações sobre a proteção de dados na Europa, consulte a [AWS postagem do blog Shared Responsibility Model and GDPR](#) no AWS Blog de segurança da.


Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com a Amazon GameLift ou outros Serviços da AWS usando o console, a API ou AWS os SDKs. AWS CLI Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Os dados GameLift específicos da Amazon são tratados da seguinte forma:

- As compilações de servidores de jogos e os scripts que você envia para a Amazon GameLift são armazenados no Amazon S3. Não há acesso direto do cliente a esses dados após fazer upload. Um usuário autorizado pode obter acesso temporário para fazer upload de arquivos, mas não pode visualizar nem atualizar os arquivos no Amazon S3 diretamente. Para excluir scripts e compilações, use o GameLift console da Amazon ou a API do serviço.
- Os dados de log da sessão de jogo são armazenados no Amazon S3 durante um período limitado de tempo após a conclusão da sessão de jogo. Usuários autorizados podem acessar os dados de registro baixando-os por meio de um link no GameLift console da Amazon ou por meio de chamadas para a API do serviço.
- Os dados métricos e de eventos são armazenados na Amazon GameLift e podem ser acessados por meio do GameLift console da Amazon ou por chamadas para a API do serviço. É possível recuperar os dados de frotas, instâncias, colocações em sessões de jogos, tíquetes de marcação de jogos, sessões de jogos e sessões de jogadores. Os dados também podem ser acessados por meio da Amazon CloudWatch e da CloudWatch Events.
- Os dados fornecidos pelo cliente são armazenados na Amazon. GameLift Os usuários autorizados podem acessá-los por chamadas para a API de serviço. Os dados potencialmente confidenciais podem incluir dados de jogadores, dados da sessão de jogador e da sessão de jogo (incluindo informações de conexão), dados da marcação de jogos e assim por diante.

 Note

Se você fornecer IDs de jogadores personalizados nas solicitações, espera-se que esses valores sejam UUIDs anônimos e não contenham informações de identificação dos jogadores.

Para obter mais informações sobre proteção de dados, consulte a publicação [AWS modelo de responsabilidade compartilhada e GDPR](#) no blog de segurança da AWS .

Criptografia inativa

A criptografia em repouso de dados GameLift específicos da Amazon é tratada da seguinte forma:

- As compilações e scripts do servidor de jogos são armazenados em buckets do Amazon S3 com criptografia no lado do servidor.
- Os dados fornecidos pelo cliente são armazenados GameLift na Amazon em um formato criptografado.

Criptografia em trânsito

As conexões com as GameLift APIs da Amazon são feitas por meio de uma conexão segura (SSL) e autenticadas usando o [AWS Signature versão 4](#) (ao se conectar por meio da AWS CLI ou do AWS SDK, a assinatura é feita automaticamente). A autenticação é gerenciada usando as políticas de acesso definidas pelo IAM para as credenciais de segurança usadas para fazer a conexão.

A comunicação direta entre clientes de jogos e servidores de jogos é a seguinte:

- Para servidores de jogos personalizados hospedados em GameLift recursos da Amazon, a comunicação não envolve o GameLift serviço da Amazon. A criptografia dessa comunicação é de responsabilidade do cliente. É possível usar frotas habilitadas para TLS para que os clientes do jogo autentiquem o servidor do jogo na conexão e criptografem toda a comunicação entre o cliente do jogo e o servidor do jogo.
- Para os servidores em tempo real com a geração de certificados TLS habilitada, o tráfego entre o cliente do jogo e os servidores em tempo real que usam o SDK do cliente em tempo real é criptografado em trânsito. O tráfego TCP é criptografado usando TLS 1.2 e o tráfego UDP é criptografado usando DTLS 1.2.

Privacidade do tráfego entre redes

Você pode acessar remotamente suas GameLift instâncias da Amazon com segurança. Para instâncias que usam Linux, o SSH fornece um canal de comunicação seguro para acesso remoto. Para instâncias que estão executando o Windows, use um cliente de protocolo de RDP (área de trabalho remota). Com o Amazon GameLift FleetIQ, o acesso remoto às suas instâncias usando o

Systems AWS Manager Session Manager e o Run Command é criptografado usando TLS 1.2, e as solicitações para criar uma conexão são assinadas usando SigV4. Para obter ajuda na conexão com uma GameLift instância gerenciada da Amazon, consulte [Conecte-se remotamente às instâncias da GameLift frota da Amazon](#).

Gerenciamento de identidade e acesso para a Amazon GameLift

O AWS Identity and Access Management (IAM) é um AWS service (Serviço da AWS) que ajuda a controlar o acesso aos recursos AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos da Amazon GameLift . O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciamento do acesso usando políticas](#)
- [Como a Amazon GameLift trabalha com o IAM](#)
- [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)
- [Solução de problemas de GameLift identidade e acesso da Amazon](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz na Amazon GameLift.

Usuário do serviço — Se você usa o GameLift serviço da Amazon para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais GameLift recursos da Amazon para fazer seu trabalho, você pode precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso na Amazon GameLift, consulte [Solução de problemas de GameLift identidade e acesso da Amazon](#).

Administrador de serviços — Se você é responsável pelos GameLift recursos da Amazon em sua empresa, provavelmente tem acesso total à Amazon GameLift. É seu trabalho determinar quais GameLift recursos e recursos da Amazon seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu

serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com a Amazon GameLift, consulte [Como a Amazon GameLift trabalha com o IAM](#).

Administrador do IAM — Se você for administrador do IAM, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso à Amazon GameLift. Para ver exemplos de políticas GameLift baseadas em identidade da Amazon que você pode usar no IAM, consulte. [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)

Autenticando com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. É necessário ser autenticado (fazer login na AWS) como o usuário raiz da Usuário raiz da conta da AWS, como usuário do IAM ou assumindo um perfil do IAM.

Você pode fazer login na AWS como uma identidade federada usando credenciais fornecidas por uma fonte de identidades. AWS IAM Identity Center Os usuários do IAM Identity Center, a autenticação única da empresa e as suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Quando você acessa a AWS usando a federação, está indiretamente assumindo um perfil.

É possível fazer login no AWS Management Console ou no portal de acesso da AWS dependendo do tipo de usuário que você é. Para obter mais informações sobre como fazer login na AWS, consulte [Como fazer login na conta da Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS.

Se você acessar a AWS programaticamente, a AWS fornecerá um kit de desenvolvimento de software (SDK) e uma interface da linha de comando (CLI) para você assinar criptograficamente as solicitações usando as suas credenciais. Se você não utilizar as ferramentas da AWS, deverá assinar as solicitações por conta própria. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte [Assinar solicitações de API da AWS](#) no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça mais informações de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte [Autenticação multifator](#) no GuiaAWS IAM Identity Center do usuário. [Usar a autenticação multifator \(MFA\) naAWS](#) no Guia do usuário do IAM.

Usuário root da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos os recursos e Serviços da AWS na conta. Essa identidade, denominada usuário raiz da Conta da AWS, e é acessada por login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários, inclusive os que precisam de acesso de administrador, usem a federação com um provedor de identidades para acessar Serviços da AWS usando credenciais temporárias.

Identidade federada é um usuário de seu diretório de usuários corporativos, um provedor de identidades da web, o AWS Directory Service, o diretório do Centro de Identidade ou qualquer usuário que acesse os Serviços da AWS usando credenciais fornecidas por meio de uma fonte de identidade. Quando as identidades federadas acessam Contas da AWS, elas assumem funções que fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no Centro de Identidade do IAM ou se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todas as suas Contas da AWS e aplicações. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [O que é o Centro de Identidade do IAM?](#) no Guia do usuário do AWS IAM Identity Center.

Grupos e usuários do IAM

Um [usuário do IAM](#) é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicação. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários

de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de funções. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas as funções fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um usuário do IAM \(em vez de uma função\)](#) no Guia do usuário do IAM.

Funções do IAM

Um [perfil do IAM](#) é uma identidade dentro da Conta da AWS que tem permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. É possível assumir temporariamente um perfil do IAM no AWS Management Console [alternando perfis](#). É possível assumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre métodos para o uso de perfis, consulte [Usar perfis do IAM](#) no Guia do usuário do IAM.

Os perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidades de terceiros](#) no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de permissões](#) no Guia do usuário do AWS IAM Identity Center.
- **Permissões temporárias para usuários do IAM:** um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas:** é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. As funções são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recurso para acesso entre contas,

consulte [Como as funções do IAM diferem das políticas baseadas em recurso](#) no Guia do usuário do IAM.

- Acesso entre serviços: alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
- Encaminhamento de sessões de acesso (FAS): qualquer pessoa que utilizar uma função ou usuário do IAM para realizar ações na AWS é considerada uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).
- Perfil de serviço: um perfil de serviço é um [perfil do IAM](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.
- Perfil vinculado ao serviço: um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir o perfil para executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados a serviço.
- Aplicações em execução no Amazon EC2: é possível usar um perfil do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso e armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém a função e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte [Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2](#) no Guia do usuário do IAM.

Para saber se deseja usar os perfis do IAM, consulte [Quando criar um perfil do IAM \(em vez de um usuário\)](#) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as a identidades ou recursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar AWS as políticas JSON da para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de perfis do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e funções na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma

política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recurso

Políticas baseadas em recurso são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar um principal](#) em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, perfis, usuários federados ou Serviços da AWS.

Políticas baseadas em recursos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

Amazon S3, AWS WAF e Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. Para saber mais sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS aceita tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões:** um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou a função no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais

informações sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.

- **Políticas de controle de serviço (SCPs):** SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à sua empresa. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cada Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizations e SCPs, consulte [Como os SCPs funcionam](#) no Guia do usuário do AWS Organizations.
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou da função e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos, consulte [Lógica da avaliação](#) de políticas no Guia do usuário do IAM.

Como a Amazon GameLift trabalha com o IAM

Antes de usar o IAM para gerenciar o acesso à Amazon GameLift, saiba quais recursos do IAM estão disponíveis para uso com a Amazon GameLift.

Recursos do IAM que você pode usar com a Amazon GameLift

Recurso do IAM	GameLift Suporte da Amazon
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não

Recurso do IAM	GameLift Suporte da Amazon
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados ao serviço	Não

Para ter uma visão de alto nível de como a Amazon GameLift e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWSos serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para a Amazon GameLift

É compatível com políticas baseadas em identidade	Sim
---	-----

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas.

Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elementos da política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para a Amazon GameLift

Para ver exemplos de políticas GameLift baseadas em identidade da Amazon, consulte [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)

Políticas baseadas em recursos na Amazon GameLift

Oferece suporte a políticas baseadas em recursos	Não
--	-----

Políticas baseadas em recurso são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar um principal](#) em uma política baseada em recursos. As entidades principais podem incluir contas, usuários, funções, usuários federados ou Serviços da AWS.

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recurso. Adicionar um principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando a entidade principal e o recurso estão em diferentes Contas da AWS, um administrador do IAM da conta confiável também deve conceder à entidade principal (usuário ou função) permissão para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a um principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

Ações políticas para a Amazon GameLift

Oferece suporte a ações de políticas	Sim
--------------------------------------	-----

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para obter uma lista das GameLift ações da Amazon, consulte [Ações definidas pela Amazon GameLift](#) na Referência de autorização de serviço.

As ações políticas na Amazon GameLift usam o seguinte prefixo antes da ação:

```
gamelift
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
  "gamelift:action1",  
  "gamelift:action2"  
]
```

Você também pode especificar várias ações usando caracteres-curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "gamelift:Describe*"
```

Para ver exemplos de políticas GameLift baseadas em identidade da Amazon, consulte [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)

Recursos de políticas para a Amazon GameLift

Oferece suporte a recursos de políticas Sim

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu [Nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um caractere curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*" 
```

Para obter uma lista dos tipos de GameLift recursos da Amazon e seus ARNs, consulte [Recursos definidos pela Amazon GameLift](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pela Amazon GameLift](#)

Alguns GameLift recursos da Amazon têm valores de ARN, o que permite que os recursos tenham seu acesso gerenciado usando políticas do IAM. O recurso de GameLift frota da Amazon tem um ARN com a seguinte sintaxe:

```
arn:${Partition}:gamelift:${Region}:${Account}:fleet/${FleetId}
```

Para obter mais informações sobre o formato de ARNs, consulte [Nomes de recursos da Amazon \(ARNs\)](#) na Referência geral da AWS.

Por exemplo, para especificar a frota `fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa` na instrução, use o seguinte ARN:

```
"Resource": "arn:aws:gamelift:us-west-2:123456789012:fleet/fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa"
```

Para especificar todas as frotas que pertencem a uma conta específica, use o caractere curinga (*):

```
"Resource": "arn:aws:gamelift:us-west-2:123456789012:fleet/*"
```

Para ver exemplos de políticas GameLift baseadas em identidade da Amazon, consulte [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)

Chaves de condição de política para a Amazon GameLift

Compatível com chaves de condição de política específicas do serviço Sim

Os administradores podem usar as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` (ou bloco de `Condition`) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único elemento `Condition`, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica OR. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos de política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

A AWS oferece suporte a chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as AWS chaves de condição globais da , consulte [AWSChaves de contexto de condição globais da](#) no Guia do usuário do IAM.

Para obter uma lista das chaves de GameLift condição da Amazon, consulte [Chaves de condição da Amazon GameLift](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pela Amazon GameLift](#).

Para ver exemplos de políticas GameLift baseadas em identidade da Amazon, consulte [Exemplos de políticas baseadas em identidade para a Amazon GameLift](#)

ACLs na Amazon GameLift

Oferece suporte a ACLs	Não
------------------------	-----

As listas de controle de acesso (ACLs) controlam quais principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

ABAC com a Amazon GameLift

Oferece suporte a ABAC (tags em políticas)	Sim
--	-----

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Na AWS, esses atributos são chamados de tags. É possível anexar tags a entidades do IAM (usuários ou funções) e a muitos recursos da AWS. A marcação de entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do recurso que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys` chaves de condição.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte [O que é ABAC?](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

Para obter um exemplo de política baseada em identidade que limita o acesso a um recurso com base em tags nesse recurso, consulte [Veja as GameLift frotas da Amazon com base em tags](#).

Usando credenciais temporárias com a Amazon GameLift

Oferece suporte a credenciais temporárias	Sim
---	-----

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Você está usando credenciais temporárias se faz login no AWS Management Console usando qualquer método, exceto um nome de usuário e uma senha. Por exemplo, quando você acessa a AWS usando o link de autenticação única (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna funções. Para obter mais informações sobre como alternar funções, consulte [Alternar para uma função \(console\)](#) no Guia do usuário do IAM.

Você pode criar credenciais temporárias manualmente usando a AWS CLI ou a API da AWS. Em seguida, você pode usar essas credenciais temporárias para acessar a AWS. A AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias no IAM](#).

Permissões principais entre serviços para a Amazon GameLift

Suporte para o recurso Encaminhamento de sessões de acesso (FAS)	Sim
--	-----

Quando você usa um usuário ou um perfil do IAM para executar ações na AWS, você é considerado uma entidade principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O recurso FAS utiliza as permissões da entidade principal que chama um AWS service (Serviço da AWS), combinadas às permissões do AWS service (Serviço da AWS) solicitante, para realizar solicitações para serviços downstream. As solicitações de FAS só são feitas quando um serviço recebe uma solicitação que exige interações com outros Serviços da AWS ou com recursos para serem concluídas. Nesse caso, você precisa ter permissões para executar ambas

as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).

Funções de serviço para a Amazon GameLift

Oferece suporte a perfis de serviço	Sim
-------------------------------------	-----

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

Warning

Alterar as permissões de uma função de serviço pode interromper a GameLift funcionalidade da Amazon. Edite as funções de serviço somente quando a Amazon GameLift fornecer orientação para fazer isso.

Permita que seus servidores de jogos GameLift hospedados na Amazon acessem outros AWS recursos, como uma AWS Lambda função ou um banco de dados do Amazon DynamoDB. Como os servidores de jogos são hospedados em frotas GameLift gerenciadas pela Amazon, você precisa de uma função de serviço que dê à Amazon acesso GameLift limitado aos seus outros AWS recursos. Para ter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Funções vinculadas a serviços para a Amazon GameLift

Oferece suporte a perfis vinculados ao serviço	Não
--	-----

Um perfil vinculado ao serviço é um tipo de perfil de serviço vinculado a um AWS service (Serviço da AWS). O serviço pode assumir a função de executar uma ação em seu nome. Os perfis vinculados ao serviço aparecem em sua Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculadas a serviços, consulte [AWS Serviços que funcionam com o IAM](#) no Guia do usuário do IAM.. Encontre um serviço na tabela

que inclua um Yes na coluna Funções vinculadas ao serviço. Selecione Sim para visualizar a documentação do perfil vinculado ao serviço para esse serviço.

Exemplos de políticas baseadas em identidade para a Amazon GameLift

Por padrão, usuários e funções não têm permissão para criar ou modificar GameLift recursos da Amazon. Eles também não podem executar tarefas usando o AWS Management Console, a AWS Command Line Interface (AWS CLI) ou a API AWS. Para conceder aos usuários permissões para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte [Criação de políticas do IAM](#) no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pela Amazon GameLift, incluindo o formato dos ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para a Amazon GameLift](#) na Referência de Autorização de Serviço.

Tópicos

- [Práticas recomendadas de políticas](#)
- [Usando o GameLift console da Amazon](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)
- [Permitir acesso de jogador para sessões de jogos](#)
- [Permitir acesso a uma GameLift fila da Amazon](#)
- [Veja as GameLift frotas da Amazon com base em tags](#)
- [Acesse um arquivo de compilação do jogo no Amazon S3](#)

Práticas recomendadas de políticas

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir GameLift recursos da Amazon em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas gerenciadas pela AWS e avance para as permissões de privilégio mínimo: para começar a conceder permissões a seus usuários e workloads, use as políticas

gerenciadas pela AWS que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis na sua Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo cliente da AWS específicas para seus casos de uso. Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso a ações de serviço, se elas forem usadas por meio de um AWS service (Serviço da AWS) específico, como o AWS CloudFormation. Para obter mais informações, consulte [Elementos de política JSON do IAM: Condition](#) no Manual do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de 100 verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário do IAM.
- Exigir autenticação multifator (MFA): se houver um cenário que exija usuários do IAM ou um usuário raiz em sua Conta da AWS, ative a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do usuário do IAM.

Para obter mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o GameLift console da Amazon

Para acessar o GameLift console da Amazon, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os GameLift recursos da

Amazon em sua Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Para garantir que essas entidades ainda possam usar o GameLift console da Amazon, adicione permissões a usuários e grupos com a sintaxe nos exemplos a seguir e em [Exemplos de permissões de administrador](#). Para ter mais informações, consulte [Gerenciar permissões de usuário para a Amazon GameLift](#).

Os usuários que trabalham com a Amazon GameLift por meio de AWS CLI AWS nossas operações de API não precisam de permissões mínimas do console. Em vez disso, é possível limitar o acesso somente às operações que o usuário precisa executar. Por exemplo, um usuário jogador, agindo em nome de clientes do jogo, exige acesso para solicitar sessões de jogo, colocar jogadores em jogos e outras tarefas.

Para obter informações sobre as permissões necessárias para usar todos os recursos GameLift do console da Amazon, consulte a sintaxe de permissões para administradores em [Exemplos de permissões de administrador](#)

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a API da AWS CLI ou da AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
  ],
}
```

```
{
  "Sid": "NavigateInConsole",
  "Effect": "Allow",
  "Action": [
    "iam:GetGroupPolicy",
    "iam:GetPolicyVersion",
    "iam:GetPolicy",
    "iam:ListAttachedGroupPolicies",
    "iam:ListGroupPolicies",
    "iam:ListPolicyVersions",
    "iam:ListPolicies",
    "iam:ListUsers"
  ],
  "Resource": "*"
}
```

Permitir acesso de jogador para sessões de jogos

Para colocar jogadores em sessões de jogo, os clientes de jogos e os serviços de back-end precisam de permissões. Para obter exemplos de políticas para esses cenários, consulte [Exemplos de permissão de usuário do jogador](#).

Permitir acesso a uma GameLift fila da Amazon

O exemplo a seguir fornece ao usuário acesso a uma GameLift fila específica da Amazon.

Essa política concede ao usuário permissões para adicionar, atualizar e excluir destinos de fila com as seguintes ações: `gamelift:UpdateGameSessionQueue`, `gamelift>DeleteGameSessionQueue` e `gamelift:DescribeGameSessionQueues`. Conforme mostrado, essa política usa o elemento `Resource` para limitar o acesso a uma única fila: `gamesessionqueue/examplequeue123`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewSpecificQueueInfo",
      "Effect": "Allow",
      "Action": [
        "gamelift:DescribeGameSessionQueues"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "arn:aws:gamelift::gamesessionqueue/examplequeue123"
  },
  {
    "Sid": "ManageSpecificQueue",
    "Effect": "Allow",
    "Action": [
      "gamelift:UpdateGameSessionQueue",
      "gamelift>DeleteGameSessionQueue"
    ],
    "Resource": "arn:aws:gamelift::gamesessionqueue/examplequeue123"
  }
]
}

```

Veja as GameLift frotas da Amazon com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos GameLift recursos da Amazon com base em tags. Este exemplo mostra como você pode criar uma política que permita visualizar uma frota se a tag `Owner` corresponder ao nome do usuário. Essa política também concede as permissões necessárias concluir essa operação no console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListFleetsInConsole",
      "Effect": "Allow",
      "Action": "gamelift:ListFleets",
      "Resource": "*"
    },
    {
      "Sid": "ViewFleetIfOwner",
      "Effect": "Allow",
      "Action": "gamelift:DescribeFleetAttributes",
      "Resource": "arn:aws:gamelift:*:*:fleet/*",
      "Condition": {
        "StringEquals": {"gamelift:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

Acesse um arquivo de compilação do jogo no Amazon S3

Depois de integrar seu servidor de jogos com a Amazon GameLift, faça o upload dos arquivos de compilação para o Amazon S3. Para GameLift que a Amazon acesse os arquivos de compilação, use a seguinte política.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::bucket-name/object-name"
    }
  ]
}
```

Para obter mais informações sobre o upload de arquivos de GameLift jogos da Amazon, consulte [Fazer upload de uma compilação de servidor personalizada para o Amazon GameLift](#).

Solução de problemas de GameLift identidade e acesso da Amazon

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com a Amazon GameLift e AWS Identity and Access Management (IAM).

Tópicos

- [Não estou autorizado a realizar uma ação na Amazon GameLift](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas de fora da minha Conta da AWS acessem meus GameLift recursos da Amazon](#)

Não estou autorizado a realizar uma ação na Amazon GameLift

Se o AWS Management Console informar que você não está autorizado a executar uma ação, entre em contato com o administrador da conta da AWS para obter assistência. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O erro de exemplo a seguir ocorre quando o usuário `mateojackson` do IAM tenta usar o console para visualizar detalhes sobre uma fila, mas não tem as permissões `gamelift:DescribeGameSessionQueues`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
gamelift:DescribeGameSessionQueues on resource: examplequeue123
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso de leitura ao recurso `examplequeue123` usando a ação `gamelift:DescribeGameSessionQueues`.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para a Amazon GameLift.

Alguns Serviços da AWS permitem que você passe um perfil existente para o serviço, em vez de criar um novo perfil de serviço ou perfil vinculado ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para realizar uma ação na Amazon GameLift. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar a função para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se você precisar de ajuda, entre em contato com seu administrador da AWS. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas de fora da minha Conta da AWS acessem meus GameLift recursos da Amazon

É possível criar um perfil que os usuários de outras contas ou pessoas fora da sua organização possam utilizar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se a Amazon GameLift oferece suporte a esses recursos, consulte [Como a Amazon GameLift trabalha com o IAM](#).
- Para saber como conceder acesso a seus atributos em todas as Contas da AWS pertencentes a você, consulte [Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você](#) no Guia de usuário do IAM.
- Para saber como conceder acesso a seus recursos para terceiros Contas da AWS, consulte [Fornecimento de acesso a Contas da AWS pertencentes a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

Registro em log e monitoramento com o Amazon GameLift

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do Amazon GameLift e das soluções da AWS. Você deve coletar dados de monitoramento de todas as partes de sua solução da AWS, para facilitar a depuração de uma falha multipontos, caso ocorra.

A AWS e o Amazon GameLift fornecem várias ferramentas para monitorar os recursos de hospedagem de jogos e para responder a possíveis incidentes.

Alarmes do Amazon CloudWatch

Com o uso de alarmes do Amazon CloudWatch, você observa uma única métrica durante um período especificado. Se a métrica exceder determinado limite, uma notificação será enviada para um tópico do Amazon SNS ou para uma política do Auto Scaling da AWS. Os alarmes do CloudWatch são acionados quando o estado deles muda e é mantido por um número especificado de períodos, não por estarem em um estado específico. Para obter mais informações, consulte [Monitorar o Amazon GameLift com o Amazon CloudWatch](#).

Logs do CloudTrail da AWS

O CloudTrail fornece um registro de ações executadas por um usuário, uma função ou um serviço da AWS no Amazon GameLift. Usando as informações coletadas pelo CloudTrail, é possível determinar a solicitação feita ao Amazon GameLift, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e outros detalhes. Para obter mais informações, consulte [Log de chamadas de API do Amazon GameLift com o AWS CloudTrail](#).

Validação de conformidade para a Amazon GameLift

A Amazon não GameLift está no escopo de nenhum programa de AWS conformidade.

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

Note

Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte [Referência dos Serviços Qualificados pela HIPAA](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços com suporte e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.
- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

Resiliência na Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [infraestrutura AWS global](#).

Além da infraestrutura AWS global, a Amazon GameLift oferece os seguintes recursos para ajudar a atender às suas necessidades de resiliência de dados:

- Filas multirregionais — As filas de sessão de GameLift jogos da Amazon são usadas para colocar novas sessões de jogo com os recursos de hospedagem disponíveis. As filas que abrangem várias regiões podem redirecionar colocações em sessões de jogos no caso de uma interrupção regional. Para obter mais informações e as melhores práticas sobre a criação de filas de sessões de jogos, consulte [Criar uma fila de sessões de jogo](#).
- Escalabilidade automática de capacidade — Mantenha a integridade e a disponibilidade de seus recursos de hospedagem usando as ferramentas de GameLift escalabilidade da Amazon. Essas ferramentas oferecem uma variedade de opções que permitem ajustar a capacidade da frota de acordo com as necessidades do jogo e dos jogadores. Para ter mais informações sobre a escalabilidade, consulte [Escalar a capacidade de hospedagem do Amazon GameLift](#).
- Distribuição entre instâncias — A Amazon GameLift distribui o tráfego de entrada em várias instâncias, dependendo do tamanho da frota. Como melhor prática, os jogos em produção devem ter várias instâncias para manter a disponibilidade caso uma instância não esteja íntegra ou não responda.
- Armazenamento Amazon S3 — As compilações e scripts de servidores de jogos que são carregados para a Amazon GameLift são armazenados no Amazon S3 usando a classe de armazenamento Standard, que usa várias replicações de data center para aumentar a resiliência. Os logs de sessão de jogo também são armazenados no Amazon S3 usando a classe de armazenamento Standard.

Segurança da infraestrutura na Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

Como um serviço gerenciado, a Amazon GameLift é protegida pelos procedimentos AWS globais de segurança de rede descritos no whitepaper [Amazon Web Services: Visão geral dos processos de segurança](#).

Você usa chamadas de API AWS publicadas para acessar a Amazon GameLift pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Recomendamos usar o TLS 1.3 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores é compatível com esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

O GameLift serviço da Amazon coloca todas as frotas nas nuvens privadas virtuais (VPCs) da Amazon para que cada frota exista em uma área logicamente isolada na nuvem. AWS Você pode usar GameLift as políticas da Amazon para controlar o acesso de VPC endpoints específicos ou VPCs específicas. Efetivamente, isso isola o acesso à rede a um determinado GameLift recurso da Amazon somente da VPC específica dentro da AWS rede. Ao criar uma frota, você especifica um intervalo de números de porta e endereços IP. Esses intervalos limitam como o tráfego de entrada pode acessar servidores de jogos hospedados em uma VPC da frota. Use as melhores práticas de segurança padrão ao selecionar as configurações de acesso à frota.

Análise de configuração e vulnerabilidade na Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

A configuração e os controles de TI são uma responsabilidade compartilhada entre a AWS e você, nosso cliente. Para obter mais informações, consulte o [modelo de responsabilidade AWS compartilhada](#). AWS lida com tarefas básicas de segurança, como sistema operacional (SO) convidado e aplicação de patches em bancos de dados, configuração de firewall e recuperação de

desastres. Esses procedimentos foram revisados e certificados por terceiros certificados. Para obter mais detalhes, consulte o seguinte recurso: [Amazon Web Services: visão geral dos processos de segurança](#) (whitepaper).

As seguintes melhores práticas de segurança também abordam a configuração e a análise de vulnerabilidades na Amazon GameLift:

- Os clientes são responsáveis pelo gerenciamento do software implantado nas GameLift instâncias da Amazon para hospedagem de jogos. Especificamente:
 - O software do aplicativo de servidor de jogos fornecido pelo cliente deve ser mantido, incluindo atualizações e patches de segurança. Para atualizar o software do servidor de jogos, faça o upload de uma nova versão para a Amazon GameLift, crie uma nova frota para ela e redirecione o tráfego para a nova frota.
 - A imagem de máquina da Amazon (AMI) base, que inclui o sistema operacional, é atualizada somente quando uma frota é criada. Para aplicar patches, atualizar e proteger o sistema operacional e outros aplicativos que fazem parte da AMI, recicle as frotas regularmente, independentemente das atualizações do servidor de jogos.
- Os clientes devem considerar a atualização regular de seus jogos com as versões mais recentes do SDK, incluindo o AWS SDK, o Amazon GameLift Server SDK e o Amazon GameLift Client SDK para servidores em tempo real.

Melhores práticas de segurança para a Amazon GameLift

Se você estiver usando o Amazon GameLift FleetIQ como um recurso independente com o Amazon EC2, consulte Segurança [no Amazon EC2 no Guia do usuário do Amazon EC2](#).

GameLift A Amazon fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Não abra portas para a Internet

É altamente recomendável não abrir portas para a Internet, pois isso representa um risco à segurança. Por exemplo, se você usa [UpdateFleetPortSettings](#) para abrir uma porta de desktop remoto como esta:

```
{
  "FleetId": "<fleet identifier>",
  "InboundPermissionAuthorizations": [
    {
      "FromPort": 3389,
      "IpRange": "0.0.0.0/0",
      "Protocol": "RDP",
      "ToPort": 3389
    }
  ]
}
```

então você está permitindo que qualquer pessoa na Internet acesse a instância.

Em vez disso, abra a porta com um endereço IP específico ou um intervalo de endereços. Por exemplo, assim:

```
{
  "FleetId": "<fleet identifier>",
  "InboundPermissionAuthorizations": [
    {
      "FromPort": 3389,
      "IpRange": "54.186.139.221/32",
      "Protocol": "TCP",
      "ToPort": 3389
    }
  ]
}
```

Saiba mais

Para obter mais informações sobre como você pode tornar seu uso da Amazon GameLift mais seguro, consulte o [pilar AWS Well-Architected Tool Segurança](#).

Guias de referência do Amazon GameLift

Esta seção contém documentação de referência sobre como usar o Amazon GameLift.

Tópicos

- [Referência de API do serviço Amazon GameLift \(SDK da AWS\)](#)
- [Referência dos Servidores em Tempo Real do Amazon GameLift](#)
- [Referência do SDK do servidor do Amazon GameLift](#)
- [Eventos de posicionamento de sessões de jogo](#)

Referência de API do serviço Amazon GameLift (SDK da AWS)

Este tópico fornece uma lista baseada em tarefas de operações de API para uso com as soluções de hospedagem gerenciada do Amazon GameLift, incluindo hospedagem para servidores de jogos personalizados e servidores em tempo real. Essas operações são empacotadas no SDK da AWS no namespace. `aws.gamelift`. [Baixe o SDK da AWS](#) ou [veja a documentação de referência da API do Amazon GameLift](#).

A API inclui dois conjuntos de operações para hospedagem gerenciada de jogos:

- [Configurar e gerenciar os recursos de hospedagem do Amazon GameLift](#)
- [Inicie sessões de jogo e junte-se aos jogadores](#)

A API do serviço Amazon GameLift também contém operações para uso com outras ferramentas e soluções do Amazon GameLift. Para obter uma lista das APIs do FleetIQ, consulte [Ações da API do FleetIQ](#). Para obter uma lista de APIs do FlexMatch para criação de jogos, consulte [Ações da API do FlexMatch](#).

Configurar e gerenciar os recursos de hospedagem do Amazon GameLift

Chame essas operações para configurar recursos de hospedagem para seus servidores de jogos, escalar a capacidade para atender à demanda dos jogadores, acessar métricas de desempenho e utilização e muito mais. Essas operações de API são usadas com servidores de jogos hospedados no Amazon GameLift, incluindo servidores em tempo real. É possível usar o [console do Amazon GameLift](#) para a maioria das tarefas de gerenciamento de recursos ou fazer chamadas para o serviço usando a ferramenta AWS Command Line Interface (AWS CLI) ou o SDK da AWS.

Preparare servidores de jogos para implantação

Carregue e configure o código do servidor de jogos do jogo em preparação para implantação e lançamento em recursos de hospedagem.

Gerenciar compilações do servidor de jogos personalizado

- [upload-build](#): carregue arquivos de compilação em um caminho local e crie um novo recurso de compilação do Amazon GameLift. Essa operação, disponível apenas como um comando da AWS CLI, é o método mais comum para carregar compilações de servidores de jogos.
- [CreateBuild](#): crie uma nova compilação usando os arquivos armazenados em um bucket do Amazon S3.
- [ListBuilds](#): obtenha uma lista de todas as compilações carregadas em uma região do Amazon GameLift.
- [DescribeBuild](#): recupere as informações associadas a uma compilação.
- [UpdateBuild](#): altere metadados da compilação, incluindo o nome e a versão da compilação.
- [DeleteBuild](#): Remova uma compilação do Amazon GameLift.

Gerenciar scripts de configuração de servidores em tempo real

- [CreateScript](#): carregue arquivos JavaScript e crie um novo recurso de script do Amazon GameLift.
- [ListScripts](#): obtenha uma lista de todos os scripts em tempo real carregados em uma região do Amazon GameLift.
- [DescribeScript](#): recupere as informações associadas a um script em tempo real.
- [UpdateScript](#): altere metadados do script e carregue o conteúdo revisado do script.
- [DeleteScript](#): remova um script em tempo real do Amazon GameLift.

Configurar recursos de computação para hospedagem

Configure recursos de hospedagem e implante-os com a compilação do servidor de jogos ou o script de configuração do em tempo real.

Criar e gerenciar frotas

- [CreateFleet](#): configure e implante uma nova frota de recursos de computação do Amazon GameLift para execução dos servidores de jogos. Depois de implantados, os servidores de jogos são automaticamente lançados conforme configurados e prontos para hospedar sessões de jogo.

- [ListFleets](#): obtenha uma lista de todas as frotas em uma região do Amazon GameLift.
- [DeleteFleet](#): encerre uma frota que não esteja mais executando servidores de jogos ou hospedando jogadores.
- Visualize/atualize as localizações da frota.
 - [CreateFleetLocations](#): adicione locais remotos a uma frota existente que suporta vários locais
 - [DescribeFleetLocationAttributes](#): obtenha uma lista de todos os locais remotos de uma frota e veja o status atual de cada local.
 - [DeleteFleetLocations](#): remova locais remotos de uma frota que suporte vários locais.
- Visualize/atualize as configurações de frota.
 - [DescribeFleetAttributes](#) / [UpdateFleetAttributes](#): visualize ou altere os metadados e as configurações de uma frota para proteção da sessão de jogo e limitação da criação de recursos.
 - [DescribeFleetPortSettings/UpdateFleetPortSettings](#): visualize ou altere as permissões de entrada (intervalos de configuração de porta e endereço IP) permitidas para uma frota.
 - [DescribeRuntimeConfiguration](#) / [UpdateRuntimeConfiguration](#): visualize ou altere quais (e quantos) processos do servidor serão executados em cada instância em uma frota.

Gerenciar capacidade da frota

- [DescribeEC2InstanceLimits](#): recupere o número máximo de instâncias permitidas para a conta atual da AWS e o nível atual de uso.
- [DescribeFleetCapacity](#): recupere a configuração atual da capacidade da frota.
- [DescribeFleetLocationCapacity](#): recupere as configurações de capacidade atuais para cada local de uma frota com vários locais.
- [UpdateFleetCapacity](#): ajuste manualmente a configuração de capacidade da frota.
- Configure o ajuste de escala automático:
 - [PutScalingPolicy](#): ative o ajuste de escala automático baseado no destino, crie uma política de ajuste de escala automático personalizado ou atualize uma política existente.
 - [DescribeScalingPolicies](#): recupere uma política de ajuste de escala automático existente.
 - [DeleteScalingPolicy](#): exclua uma política de ajuste de escala automático e impeça que ela afete a capacidade de uma frota.
 - [StartFleetActions](#): reinicie as políticas de ajuste de escala automático de uma frota.
 - [StopFleetActions](#): suspenda as políticas de ajuste de escala automático de uma frota.

Monitore a atividade da frota.

- [DescribeFleetUtilization](#): recupere as estatísticas do número de processos do servidor, sessões de jogos e jogadores que estiverem ativos no momento em uma frota.
- [DescribeFleetLocationUtilization](#): recupere estatísticas de utilização para cada local em uma frota com vários locais.
- [DescribeFleetEvents](#): visualize os eventos registrados para uma frota durante um período especificado.
- [DescribeGameSessions](#): recupere os metadados da sessão de jogo, incluindo o tempo de execução de um jogo e a contagem atual de jogadores.

Configurar filas para o posicionamento ideal de sessão de jogo

Configure as filas de várias frotas e várias regiões para posicionar sessões de jogo com os melhores recursos de hospedagem disponíveis para custo, latência e resiliência.

- [CreateGameSessionQueue](#): crie uma fila para usar no processo de solicitação de posicionamento de sessões de jogo.
- [DescribeGameSessionQueues](#): recupere as filas de sessões de jogo definidas em uma região do Amazon GameLift.
- [UpdateGameSessionQueue](#): altere a configuração de uma fila de sessões de jogo.
- [DeleteGameSessionQueue](#): remova uma fila de sessões de jogo da região.

Gerenciar aliases

Use aliases para representar suas frotas ou crie um destino alternativo de terminal. Os aliases são úteis ao fazer a transição da atividade do jogo de uma frota para outra, como durante as atualizações de compilação do servidor de jogos.

- [CreateAlias](#): defina um novo alias e, opcionalmente, atribua-o a uma frota.
- [ListAliases](#): obtenha todos os aliases de frota definidos em uma região do Amazon GameLift.
- [DescribeAlias](#): recupere as informações de um alias existente.
- [UpdateAlias](#): altere as configurações de um alias, como o encaminhamento dele de uma frota para outra.
- [DeleteAlias](#): remova um alias da região.

- [ResolveAlias](#): obtenha o ID da frota para o qual um alias especificado está apontando.

Acessar instâncias de hospedagem

Visualize informações sobre instâncias individuais em uma frota ou solicite acesso remoto a uma instância de frota especificada para solução de problemas.

- [DescribeInstances](#): obtenha as informações de cada instância em uma frota, incluindo o ID, o endereço IP, o local e o status dela.
- [GetInstanceAccess](#): solicite as credenciais de acesso necessárias para se conectar remotamente a uma instância especificada em uma frota.

Configurar o emparelhamento da VPC

Crie e gerencie conexões de emparelhamento da VPC entre os recursos de hospedagem do Amazon GameLift e outros recursos da AWS.

- [CreateVpcPeeringAuthorization](#): autorize uma conexão de emparelhamento com uma das VPCs.
- [DescribeVpcPeeringAuthorizations](#): recupere as autorizações válidas da conexão de emparelhamento.
- [DeleteVpcPeeringAuthorization](#): exclua uma autorização de conexão de emparelhamento.
- [CreateVpcPeeringConnection](#): estabeleça uma conexão de emparelhamento entre a VPC de uma frota do Amazon GameLift e uma das VPCs.
- [DescribeVpcPeeringConnections](#): recupere as informações de conexões de emparelhamento da VPC ativas ou pendentes com uma frota do Amazon GameLift.
- [DeleteVpcPeeringConnection](#): exclua uma conexão de emparelhamento da VPC com uma frota do Amazon GameLift.

Inicie sessões de jogo e junte-se aos jogadores

Chame essas operações do serviço de cliente do jogo para iniciar novas sessões de jogo, obter informações sobre sessões de jogo existentes e adicionar jogadores às sessões de jogo. Essas operações são destinadas para uso com servidores de jogos personalizados hospedados no Amazon GameLift. Se estiver usando servidores em tempo real, gerencie as sessões de jogo usando o [Referência de API \(C#\) do cliente dos servidores em tempo real](#).

- Inicie novas sessões de jogo para um ou mais jogadores.
 - [StartGameSessionPlacement](#): peça ao Amazon GameLift para encontrar os melhores recursos de hospedagem disponíveis e iniciar uma nova sessão de jogo. Esse é o método preferido para criar novas sessões de jogo. Ele se baseia em filas de sessões de jogo para rastrear a disponibilidade de hospedagem em várias regiões e usa algoritmos FlectiQ para priorizar posicionamentos com base na latência do jogador, custo de hospedagem, localização etc.
 - [DescribeGameSessionPlacement](#): obtenha detalhes e status sobre uma solicitação de posicionamento.
 - [StopGameSessionPlacement](#): cancele uma solicitação de posicionamento.
 - [CreateGameSession](#): inicie uma nova sessão de jogo em uma frota específica. Essa operação oferece maior controle sobre onde iniciar a sessão do jogo, em vez de usar o FlectiQ para avaliar as opções de posicionamento. Você deve adicionar jogadores à nova sessão de jogo em uma etapa separada.
- Coloque jogadores em sessões de jogo existentes. Encontre sessões de jogo em execução com slots de jogadores disponíveis e reserve-os para novos jogadores.
 - [CreatePlayerSession](#): reserve um slot aberto para um jogador ingressar em uma sessão de jogo.
 - [CreatePlayerSessions](#): reserve slots abertos para vários jogadores ingressarem em uma sessão de jogo.
- Trabalhe com os dados da sessão de jogo e da sessão de jogadores. Gerencie informações sobre sessões de jogo e sessões de jogadores.
 - [SearchGameSessions](#): solicite uma lista de sessões de jogo ativas com base em um conjunto de critérios de pesquisa.
 - [DescribeGameSessions](#): recupere os metadados de uma ou mais sessões de jogos, incluindo o tempo de atividade e a contagem atual de jogadores.
 - [DescribeGameSessionDetails](#): recupere os metadados, incluindo e a configuração de proteção da sessão de jogo de uma ou mais sessões de jogo.
 - [DescribePlayerSessions](#): obtenha detalhes sobre a atividade do jogador, incluindo status, tempo de jogo e dados do jogador.
 - [UpdateGameSession](#): altere as configurações da sessão de jogo, como a contagem máxima de jogadores e a política de participação.
 - [GetGameSessionLogUrl](#): obtenha o local dos logs salvos de uma sessão de jogo.

Referência dos Servidores em Tempo Real do Amazon GameLift

Esta seção contém documentação de referência sobre o SDK dos Servidores em Tempo Real do Amazon GameLift. Ela inclui a API de cliente em tempo real, bem como orientações para configurar seu script dos servidores em tempo real.

Tópicos

- [Referência de API \(C#\) do cliente dos servidores em tempo real](#)
- [Referência de script de Servidores em Tempo Real do Amazon GameLift](#)

Referência de API (C#) do cliente dos servidores em tempo real

Use a API do cliente em tempo real para preparar os clientes de jogos multijogador para uso com os Servidores em Tempo Real do Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Preparar seu servidor em tempo real](#). A API do cliente contém um conjunto de chamadas de API síncronas e retornos de chamada assíncronos que permitem que um cliente do jogo se conecte a um servidor em tempo real e troque mensagens e dados com outros clientes de jogos por meio do servidor.

A API é definida nas seguintes bibliotecas:

Client.cs

- [Ações síncronas](#)
- [Retornos de chamada assíncronos](#)
- [Tipos de dados](#)

Para configurar a API do cliente em tempo real

1. Baixe o [SDK do cliente em tempo real do Amazon GameLift](#).
2. Compile as bibliotecas do C# SDK. Localize o arquivo de solução `GameLiftRealtimeClientSdkNet45.sln`. Consulte o arquivo do `README.md` para o SDK de servidor para obter os requisitos mínimos e as opções de compilação adicionais. Em um IDE, carregue o arquivo da solução. Para gerar as bibliotecas do SDK, restaure os pacotes do NuGet e compile a solução.
3. Adicione as bibliotecas de clientes em tempo real ao projeto do cliente do jogo.

Referência da API (C#) de clientes de servidores em tempo real: ações

Essa referência de API C# de clientes em tempo real pode ajudar a preparar o jogo multijogador a ser usado com os servidores em tempo real implantados em frotas do Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Preparar seu servidor em tempo real](#).

- [Ações síncronas](#)
- [Retornos de chamada assíncronos](#)
- [Tipos de dados](#)

Client()

Inicializa um novo cliente para se comunicar com o servidor em tempo real e identifica o tipo de conexão a ser usado.

Sintaxe

```
public Client(ClientConfiguration configuration)
```

Parâmetros

clientConfiguration

Detalhes de configuração especificando o tipo de conexão cliente/servidor. Você pode optar por chamar Client() sem esse parâmetro; no entanto, essa abordagem resulta em uma conexão não segura por padrão.

Digite: [ClientConfiguration](#)

Obrigatório: não

Valor de retorno

Retorna uma instância do cliente Realtime para uso com a comunicação com o servidor Realtime.

Connect()

Solicita uma conexão com um processo de servidor que está hospedando uma sessão do jogo.

Sintaxe

```
public ConnectionStatus Connect(string endpoint, int remoteTcpPort, int listenPort,
    ConnectionToken token)
```

Parâmetros

endpoint

Nome DNS ou endereço IP da sessão de jogo à qual se conectar. O endpoint é especificado em um objeto `GameSession`, que é retornado em resposta a uma chamada do cliente das ações API do Amazon GameLift do SDK da AWS [StartGameSessionPlacement](#), [CreateGameSession](#) ou [DescribeGameSessions](#).

Note

Se o servidor Realtime estiver em execução em uma frota com um certificado TLS, você deverá usar o nome DNS.

Tipo: string

Obrigatório: sim

remoteTcpPort

Número da porta da conexão TCP atribuída à sessão do jogo. Essas informações são especificadas em um objeto `GameSession`, que é retornado em resposta a uma solicitação [StartGameSessionPlacement](#) [CreateGameSession](#) ou [DescribeGameSession](#).

Tipo: inteiro

Valores válidos: 1900 a 2000.

Obrigatório: sim

listenPort

Número da porta em que o cliente de jogo está escutando mensagens enviadas usando o canal UDP.

Tipo: inteiro

Valores válidos: 33400 a 33500.

Obrigatório: sim

token

Informações opcionais que identificam o cliente do jogo solicitante para o processo do servidor.

Digite: [ConnectionToken](#)

Obrigatório: sim

Valor de retorno

Retorna um valor de enumeração [ConnectionStatus](#) indicando o status da conexão do cliente.

Disconnect()

Quando conectado a uma sessão de jogo, desconecta o cliente do jogo da sessão do jogo.

Sintaxe

```
public void Disconnect()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Esse método não retorna nada.

NewMessage()

Cria um novo objeto de mensagem com um código de operação especificado. Assim que um objeto de mensagem for retornado, conclua o conteúdo da mensagem especificando um destino, atualizando o método de entrega e adicionando uma carga útil de dados conforme necessário. Uma vez concluído, envie a mensagem usando `SendMessage()`.

Sintaxe

```
public RTMessage NewMessage(int opCode)
```


Parâmetros

opCode

Código de operação definido pelo desenvolvedor que identifica um evento de jogo ou ação, como uma movimentação do jogador ou uma notificação do servidor.

Tipo: inteiro

Obrigatório: sim

Valor de retorno

Retorna um objeto [RTMessage](#) que contém o código de operação especificado e o método de entrega padrão. O parâmetro de intenção de entrega é definido como FAST por padrão.

SendMessage()

Envia uma mensagem para um jogador ou grupo usando o método de entrega especificado.

Sintaxe

```
public void SendMessage(RTMessage message)
```

Parâmetros

mensagem

Objeto de mensagem que especifica o destinatário o método de entrega e o conteúdo da mensagem.

Digite: [RTMessage](#)

Obrigatório: sim

Valor de retorno

Esse método não retorna nada.

JoinGroup()

Adiciona o jogador à associação de um grupo especificado. Os grupos podem conter qualquer um dos jogadores conectados ao jogo. Uma vez associado, o jogador recebe todas as futuras mensagens enviadas ao grupo e pode enviar mensagens para todo o grupo.

Sintaxe

```
public void JoinGroup(int targetGroup)
```

Parâmetros

targetGroup

ID exclusivo que identifica o grupo ao qual adicionar o jogador. Os IDs de grupo são definidos pelo desenvolvedor.

Tipo: inteiro

Obrigatório: sim

Valor de retorno

Esse método não retorna nada. Como essa solicitação é enviada usando o método de entrega confiável (TCP), uma solicitação com falha aciona o retorno de chamada [OnError\(\)](#).

LeaveGroup()

Remove o jogador da associação de um grupo especificado. Assim que não estiver mais no grupo, o jogador não receberá as mensagens enviadas ao grupo e não poderá enviar mensagens para todo o grupo.

Sintaxe

```
public void LeaveGroup(int targetGroup)
```

Parâmetros

targetGroup

ID exclusivo que identifica o grupo do qual remover o jogador. Os IDs de grupo são definidos pelo desenvolvedor.

Tipo: inteiro

Obrigatório: sim

Valor de retorno

Esse método não retorna nada. Como essa solicitação é enviada usando o método de entrega confiável (TCP), uma solicitação com falha aciona o retorno de chamada [OnError\(\)](#).

RequestGroupMembership()

Solicita que uma lista de jogadores no grupo especificado seja enviada para o cliente do jogo. Qualquer jogador pode solicitar essas informações, independentemente de ele ser um membro do grupo ou não. Em resposta a essa solicitação, a lista de associações é enviada ao cliente por meio de um retorno de chamada [OnGroupMembershipUpdated\(\)](#).

Sintaxe

```
public void RequestGroupMembership(int targetGroup)
```

Parâmetros

targetGroup

ID exclusivo que identifica o grupo do qual obter informações de associação. Os IDs de grupo são definidos pelo desenvolvedor.

Tipo: inteiro

Obrigatório: sim

Valor de retorno

Esse método não retorna nada.

Referência da API (C#) do cliente dos servidores em tempo real: retornos de chamada assíncronos

Use essa referência de API C# do cliente em tempo real para ajudar a preparar o jogo multijogador a ser usado com os servidores em tempo real implantados em frotas do Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Preparar seu servidor em tempo real](#).

- [Ações síncronas](#)
- Retornos de chamada assíncronos
- [Tipos de dados](#)

Um cliente de jogo precisa implementar esses métodos de retorno de chamada para responder a eventos. O servidor em tempo real chama esses retornos de chamada para enviar informações relacionadas ao jogo para o cliente do jogo. Retornos de chamada dos mesmos eventos também podem ser implementados com a lógica de jogos personalizada no script de servidor em tempo real. Consulte [Retornos de chamada de script para servidores em tempo real](#).

Os métodos de retorno de chamada são definidos em `ClientEvents.cs`.

OnOpen()

Chamado quando o processo do servidor aceita a solicitação de conexão do cliente do jogo e abre uma conexão.

Sintaxe

```
public void OnOpen()
```

Parâmetros

Esse método não aceita parâmetros.

Valor de retorno

Esse método não retorna nada.

OnClose()

Chamado quando o processo do servidor encerra a conexão com o cliente do jogo, como após uma sessão do jogo.

Sintaxe

```
public void OnClose()
```

Parâmetros

Esse método não aceita parâmetros.

Valor de retorno

Esse método não retorna nada.

OnError()

Chamado quando ocorre uma falha para uma solicitação de API do cliente em tempo real. Esse retorno de chamada pode ser personalizado para lidar com uma variedade de erros de conexão.

Sintaxe

```
private void OnError(byte[] args)
```

Parâmetros

Esse método não aceita parâmetros.

Valor de retorno

Esse método não retorna nada.

OnDataReceived()

Chamado quando o cliente do jogo recebe uma mensagem do servidor em tempo real. Este é o método principal pelo qual as mensagens e as notificações são recebidas por um cliente do jogo.

Sintaxe

```
public void OnDataReceived(DataReceivedEventArgs dataReceivedEventArgs)
```

Parâmetros

dataReceivedEventArgs

Informações relacionadas à atividade de mensagem.

Digite: [DataReceivedEventArgs](#)

Obrigatório: sim

Valor de retorno

Esse método não retorna nada.

OnGroupMembershipUpdated()

Chamado quando a associação de um grupo ao qual o jogador pertence foi atualizada. Esse retorno de chamada também é invocado quando um cliente chama `RequestGroupMembership`.

Sintaxe

```
public void OnGroupMembershipUpdated(GroupMembershipEventArgs groupMembershipEventArgs)
```

Parâmetros

groupMembershipEventArgs

Informações relacionadas à atividade de associação do grupo.

Digite: [GroupMembershipEventArgs](#)

Obrigatório: sim

Valor de retorno

Esse método não retorna nada.

Referência da API (C#) do cliente dos servidores em tempo real: tipos de dados

Essa referência de API C# do cliente em tempo real pode ajudar a preparar o jogo multijogador a ser usado com os servidores em tempo real implantados em frotas do Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Preparar seu servidor em tempo real](#).

- [Ações síncronas](#)
- [Retornos de chamada assíncronos](#)
- Tipos de dados

ClientConfiguration

Informações sobre como o cliente do jogo se conecta a um servidor em tempo real.

Índice

ConnectionType

Tipo de conexão cliente/servidor a ser usada, protegida ou não protegida. Se você não especificar um tipo de conexão, o padrão será desprotegido.

Note

Ao se conectar a um servidor Realtime em uma frota protegida por um certificado TLS, você deve usar o valor `RT_OVER_WSS_DTLS_TLS12`.

Tipo: um valor `ConnectionType` [enum](#).

Obrigatório: não

ConnectionToken

Informações sobre o cliente do jogo e/ou jogador que está solicitando uma conexão com um servidor em tempo real.

Índice

PlayerSessionId

ID exclusiva emitida pelo Amazon GameLift quando uma nova sessão de jogadores é criada. [O ID da sessão de jogadores é especificado em um objeto `PlayerSession`, que é retornado em resposta a uma chamada do cliente para as ações da API GameLift `StartGameSessionPlacement`, `CreateGameSession`, `DescribeGameSessionPlacement` ou `DescribePlayerSessions`.](#)

Tipo: string

Obrigatório: sim

payload

Informações definidas pelo desenvolvedor a serem comunicadas ao servidor em tempo real na conexão. Isso inclui quaisquer dados arbitrários que possam ser usados para um mecanismo de login personalizado. Para obter exemplos, uma carga útil pode fornecer informações de

autenticação a serem processadas pelo script do servidor em tempo real antes de permitir que um cliente se conecte.

Tipo: matriz de bytes

Obrigatório: não

RTMessage

Conteúdo e informações de entrega de uma mensagem. Uma mensagem deve especificar um jogador ou um grupo de destino.

Índice

opCode

Código de operação definido pelo desenvolvedor que identifica um evento de jogo ou ação, como uma movimentação do jogador ou uma notificação do servidor. O código de operação de uma mensagem fornece o contexto para a carga útil de dados que está sendo fornecida. As mensagens que são criadas usando `NewMessage()` já têm o código de operação definido, mas podem ser alteradas a qualquer momento.

Tipo: inteiro

Obrigatório: sim

targetPlayer

ID exclusivo que identifica o jogador que é o destinatário pretendido da mensagem que está sendo enviada. O destino pode ser o próprio servidor (usando o ID do servidor) ou outro jogador (usando um ID do jogador).

Tipo: inteiro

Obrigatório: não

targetGroup

ID exclusivo que identifica o grupo que é o destinatário pretendido da mensagem que está sendo enviada. Os IDs de grupo são definidos pelo desenvolvedor.

Tipo: inteiro

Obrigatório: não

deliveryIntent

Indica se a mensagem deve ser enviada usando a conexão TCP confiável ou o canal UDP rápido. Mensagens criadas usando [NewMessage\(\)](#).

Tipo: DeliveryIntent enum

Valores válidos: FAST | RELIABLE

Obrigatório: sim

payload

Conteúdo da mensagem. Essas informações são estruturadas conforme necessário para serem processadas pelo cliente do jogo com base no código de operação fornecido. Pode conter dados de estado do jogo ou outras informações que precisem ser transmitidas entre os clientes de jogos ou entre um cliente de jogo e o servidor em tempo real.

Tipo: matriz de bytes

Obrigatório: não

DataReceivedEventArgs

Os dados fornecidos com um retorno de chamada [OnDataReceived\(\)](#).

Índice

remetente

ID exclusivo que identifica a entidade (ID do jogador ou ID do servidor) que originou a mensagem.

Tipo: inteiro

Obrigatório: sim

opCode

Código de operação definido pelo desenvolvedor que identifica um evento de jogo ou ação, como uma movimentação do jogador ou uma notificação do servidor. O código de operação de uma mensagem fornece o contexto para a carga útil de dados que está sendo fornecida.

Tipo: inteiro

Obrigatório: sim

data

Conteúdo da mensagem. Essas informações são estruturadas conforme necessário para serem processadas pelo cliente do jogo com base no código de operação fornecido. Pode conter dados de estado do jogo ou outras informações que precisem ser transmitidas entre os clientes de jogos ou entre um cliente de jogo e o servidor em tempo real.

Tipo: matriz de bytes

Obrigatório: não

GroupMembershipEventArgs

Os dados fornecidos com um retorno de chamada [OnGroupMembershipUpdated\(\)](#).

Índice

remetente

ID exclusivo que identifica o jogador que solicitou uma atualização da associação do grupo.

Tipo: inteiro

Obrigatório: sim

opCode

Código de operação definido pelo desenvolvedor que identifica um evento ou uma ação de jogo.

Tipo: inteiro

Obrigatório: sim

groupId

ID exclusivo que identifica o grupo que é o destinatário pretendido da mensagem que está sendo enviada. Os IDs de grupo são definidos pelo desenvolvedor.

Tipo: inteiro

Obrigatório: sim

playerId

Lista de IDs de jogadores que são membros atuais do grupo especificado.

Tipo: matriz de inteiros

Obrigatório: sim

Enums

As enumerações definidas para o SDK do cliente em tempo real são definidas da seguinte forma:

ConnectionStatus

- **CONNECTED**: o cliente do jogo está conectado ao servidor em tempo real com uma conexão TCP apenas. Todas as mensagens, independentemente da intenção de entrega, são enviadas via TCP.
- **CONNECTED_SEND_FAST**: o cliente do jogo está conectado ao servidor em tempo real com uma conexão TCP e uma UDP. No entanto, a capacidade de receber mensagens por meio de UDP ainda não é verificada; como resultado, todas as mensagens enviadas para o cliente do jogo usam TCP.
- **CONNECTED_SEND_AND_RECEIVE_FAST**: o cliente do jogo está conectado ao servidor em tempo real com uma conexão TCP e uma UDP. O cliente do jogo pode enviar e receber mensagens usando TCP ou UDP.
- **CONNECTING**: o cliente do jogo enviou uma solicitação de conexão e o servidor em tempo real a está processando.
- **DISCONNECTED_CLIENT_CALL**: o cliente do jogo foi desconectado do servidor em tempo real em resposta a uma [Disconnect\(\)](#) solicitação do cliente do jogo.
- **DISCONNECTED**: o cliente do jogo foi desconectado do servidor em tempo real por um motivo diferente de uma chamada de desconexão do cliente.

ConnectionType

- **RT_OVER_WSS_DTLS_TLS12**: tipo de conexão segura.

Para uso com servidores Realtime que estão em execução em uma frota do GameLift com um certificado TLS gerado. Ao usar uma conexão segura, o tráfego TCP é criptografado usando TLS 1.2 e o tráfego UDP é criptografado usando DTLS 1.2.

- **RT_OVER_WS_UDP_UNSECURED**: tipo de conexão não segura.
- **RT_OVER_WEBSOCKET**: tipo de conexão não segura. Esse valor não é mais preferível.

DeliveryIntent

- FAST: entregue usando um canal UDP.
- RELIABLE: entregue usando uma conexão TCP.

Referência de script de Servidores em Tempo Real do Amazon GameLift

Use esses recursos para criar lógica personalizada nos scripts em tempo real.

Tópicos

- [Retornos de chamada de script para servidores em tempo real](#)
- [Interface de servidores em tempo real](#)

Retornos de chamada de script para servidores em tempo real

É possível fornecer uma lógica personalizada para responder a eventos implementando esses retornos de chamada em seu script em tempo real.

Init

Inicializa o servidor em tempo real e recebe uma interface de servidor em tempo real.

Sintaxe

```
init(rtsession)
```

onMessage

Invocado quando uma mensagem recebida é enviada para o servidor.

Sintaxe

```
onMessage(gameMessage)
```

onHealthCheck

Invocada para definir o status de integridade da sessão de jogo. Por padrão, o status de integridade é íntegro (ou `true`). Esse retorno de chamada pode ser implementado para executar verificações de integridade personalizadas e retornar um status.

Sintaxe

```
onHealthCheck()
```

onStartGameSession

Invocada quando uma nova sessão de jogo é iniciada, com um objeto de sessão do jogo transmitido.

Sintaxe

```
onStartGameSession(session)
```

onProcessTerminate

Invocada quando o processo do servidor está sendo encerrado pelo serviço do Amazon GameLift. Pode atuar como um trigger para sair corretamente da sessão do jogo. Não há necessidade de chamar `processEnding()`.

Sintaxe

```
onProcessTerminate()
```

onPlayerConnect

Invocado quando um jogador solicita uma conexão e passou a validação inicial.

Sintaxe

```
onPlayerConnect(connectMessage)
```

onPlayerAccepted

Invocado quando uma conexão do jogador é aceita.

Sintaxe

```
onPlayerAccepted(player)
```

onPlayerDisconnect

Invocado quando um jogador se desconecta da sessão do jogo, enviando uma solicitação de desconexão ou por outros meios.

Sintaxe

```
onPlayerDisconnect(peerId)
```

onProcessStarted

Invocado quando um processo do servidor é iniciado. Esse retorno de chamada permite que o script personalizado execute as tarefas necessárias para se preparar para hospedar uma sessão de jogo.

Sintaxe

```
onProcessStarted(args)
```

onSendToPlayer

Invocado quando uma mensagem é recebida no servidor de um jogador para ser entregue para outro jogador. Esse processo é executado antes que a mensagem seja entregue.

Sintaxe

```
onSendToPlayer(gameMessage)
```

onSendToGroup

Invocado quando uma mensagem é recebida no servidor de um jogador para ser entregue para um grupo. Esse processo é executado antes que a mensagem seja entregue.

Sintaxe

```
onSendToGroup(gameMessage)
```

onPlayerJoinGroup

Invocado quando um jogador envia uma solicitação para ingressar em um grupo.

Sintaxe

```
onPlayerJoinGroup(groupId, peerId)
```

onPlayerLeaveGroup

Invocado quando um jogador envia uma solicitação para sair de um grupo.

Sintaxe

```
onPlayerLeaveGroup(groupId, peerId)
```

Interface de servidores em tempo real

Quando um script em tempo real é inicializado, uma interface é retornada para o servidor em tempo real. Este tópico descreve as propriedades e métodos disponíveis por meio da interface. Saiba mais sobre como gravar scripts em tempo real e visualize um script detalhado de exemplo em [Criar um script em tempo real](#).

A interface em tempo real fornece acesso aos seguintes objetos:

- session
- player
- gameMessage
- configuration

Objeto de sessão em tempo real

Use esses métodos para acessar informações relacionadas ao servidor e para executar ações relacionadas ao servidor.

```
getPlayers()
```

Recupera uma lista de IDs pares para jogadores que estão conectados no momento à sessão de jogo. Retorna uma matriz de objetos de jogador.

Sintaxe

```
rtSession.getPlayers()
```

```
broadcastGroupMembershipUpdate()
```

Aciona a entrega de uma lista atualizada de associações do grupo ao grupo de jogadores. Especifique qual associação transmitir (groupIdToBroadcast) e o grupo para receber a atualização (targetGroupId). Os IDs de grupo devem ser um número inteiro positivo ou “-1” para indicar todos os grupos. Consulte [Exemplo de script de servidores em tempo real](#) para ver um exemplo de IDs de grupo definidos pelo usuário.

Sintaxe

```
rtSession.broadcastGroupMembershipUpdate(groupIdToBroadcast, targetGroupId)
```

getServerId()

Recupera o identificador do ID par exclusivo do servidor, que é usado para rotear mensagens para o servidor.

Sintaxe

```
rtSession.getServerId()
```

getAllPlayersGroupId()

Recupera o ID de grupo para o grupo padrão que contém todos os jogadores conectados no momento à sessão de jogo.

Sintaxe

```
rtSession.getAllPlayersGroupId()
```

processEnding()

Aciona o servidor em tempo real para encerrar o servidor de jogos. Essa função deve ser chamada no script em tempo real para sair corretamente de uma sessão de jogo.

Sintaxe

```
rtSession.processEnding()
```

getGameSessionId()

Recupera o ID exclusivo da sessão do jogo em execução no momento.

Sintaxe

```
rtSession.getGameSessionId()
```


getLogger()

Recupera a interface para o registro em log. Use essa opção para registrar instruções de log que serão capturadas em seus logs de sessão de jogo. O registrador em log é compatível com o uso de instruções “info”, “warn” e “error”. Por exemplo: `logger.info("<string>")`.

Sintaxe

```
rtSession.getLogger()
```

sendMessage()

Envia uma mensagem, criada usando `newTextGameMessage` ou `newBinaryGameMessage`, do servidor em tempo real para um jogador destinatário usando o canal UDP. Identifique o destinatário usando o ID par do jogador.

Sintaxe

```
rtSession.sendMessage(gameMessage, targetPlayer)
```

sendGroupMessage()

Envia uma mensagem, criada usando `newTextGameMessage` ou `newBinaryGameMessage`, do servidor em tempo real a todos os jogadores em um grupo de jogadores usando o canal UDP. Os IDs de grupo devem ser um número inteiro positivo ou “-1” para indicar todos os grupos. Consulte [Exemplo de script de servidores em tempo real](#) para ver um exemplo de IDs de grupo definidos pelo usuário.

Sintaxe

```
rtSession.sendGroupMessage(gameMessage, targetGroup)
```

sendReliableMessage()

Envia uma mensagem, criada usando `newTextGameMessage` ou `newBinaryGameMessage`, do servidor em tempo real para um jogador destinatário usando o canal TCP. Identifique o destinatário usando o ID par do jogador.

Sintaxe

```
rtSession.sendReliableMessage(gameMessage, targetPlayer)
```

sendReliableGroupMessage()

Envia uma mensagem, criada usando `newTextGameMessage` ou `newBinaryGameMessage`, do servidor em tempo real para todos os jogadores em um grupo de jogadores usando o canal TCP. Os IDs de grupo que devem ser um número inteiro positivo ou “-1” para indicar todos os grupos. Consulte [Exemplo de script de servidores em tempo real](#) para ver um exemplo de IDs de grupo definidos pelo usuário.

Sintaxe

```
rtSession.sendReliableGroupMessage(gameMessage, targetGroup)
```

newTextGameMessage()

Cria uma nova mensagem que contém texto, a ser enviada do servidor aos jogadores destinatários usando as funções `SendMessage`. O formato da mensagem é semelhante ao formato usado no SDK do cliente Realtime (consulte [RTMessage](#)). Informa um objeto `gameMessage`.

Sintaxe

```
rtSession.newTextGameMessage(opcode, sender, payload)
```

newBinaryGameMessage()

Cria uma nova mensagem que contém dados binários a ser enviada do servidor aos jogadores destinatários usando as funções `SendMessage`. O formato da mensagem é semelhante ao formato usado no SDK do cliente Realtime (consulte [RTMessage](#)). Informa um objeto `gameMessage`.

Sintaxe

```
rtSession.newBinaryGameMessage(opcode, sender, binaryPayload)
```

Objeto do jogador

Acessar informações relacionadas ao jogador.

`player.peerId`

ID exclusivo que é atribuído a um cliente do jogo quando ele se conecta ao servidor em tempo real e ingressa na sessão do jogo.

player.playerSessionId

O ID da sessão do jogador que foi referenciado pelo cliente do jogo quando conectou-se ao servidor em tempo real e ingressou na sessão do jogo.

Objeto de mensagem de jogo

Use esses métodos para acessar as mensagens que são recebidas pelo servidor em tempo real. As mensagens recebidas de clientes de jogos têm a estrutura [RTMessage](#).

getPayloadAsText()

Obtém a carga útil da mensagem do jogo como texto.

Sintaxe

```
gameMessage.getPayloadAsText()
```

gameMessage.opcode

Código de operação contido em uma mensagem.

gameMessage.payload

Carga contida em uma mensagem. Pode ser texto ou binário.

gameMessage.sender

ID par do cliente do jogo que enviou uma mensagem.

gameMessage.reliable

Booliano indica se a mensagem foi enviada por meio de TCP (verdadeiro) ou UDP (falso).

Objeto de configuração

O objeto de configuração pode ser usado para substituir as configurações padrão.

configuration.maxPlayers

O número máximo de conexões cliente/servidor que podem ser aceitas pelo RealTimeServers.

O padrão é 32.

`configuration.pingIntervalTime`

Intervalo de tempo em milissegundos em que o servidor tentará enviar um ping a todos os clientes conectados para verificar se as conexões estão íntegras.

O padrão é 3000 ms.

Referência do SDK do servidor do Amazon GameLift

Esta seção contém documentação de referência sobre o SDK do servidor do Amazon GameLift. Use o SDK do servidor para integrar os servidores de jogos personalizados para se comunicar com o serviço Amazon GameLift.

Tópicos

- [Referência de SDK do servidor do Amazon GameLift para C++](#)
- [Referência do SDK do servidor do Amazon GameLift para C#](#)
- [Referência do SDK do servidor do Amazon GameLift para Go](#)
- [Referência do SDK do servidor do Amazon GameLift para Unreal Engine](#)

Referência de SDK do servidor do Amazon GameLift para C++

Você poderá usar essa referência do SDK do servidor C++ do Amazon GameLift para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência do SDK 5.x do servidor Amazon GameLift para C++](#)
- [Referência do SDK 3.x do servidor C++ do Amazon GameLift](#)

Referência do SDK 5.x do servidor Amazon GameLift para C++

Esta referência do 5.x do Server SDK C++ do Amazon GameLift pode ajudar a preparar o jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Note

Este tópico descreve a API C++ do Amazon GameLift que você pode usar ao criar com a Biblioteca padrão C++ (`std`). Especificamente, essa documentação se aplica ao código que você compila com a opção `-DDGAMELIFT_USE_STD=1`.

Tópicos

- [Referência 5.x GameLift do Amazon Server SDK \(C++\): Ações](#)
- [Referência GameLift do Amazon Server SDK \(C++\): tipos de dados](#)

Referência 5.x GameLift do Amazon Server SDK (C++): Ações

Você pode usar essa referência do SDK do servidor Amazon GameLift C++ para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon. GameLift Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Note

Este tópico descreve a API do Amazon GameLift C++ que você pode usar ao criar com a C++ Standard Library (`std`). Especificamente, essa documentação se aplica ao código que você compila com a opção `-DDGAMELIFT_USE_STD=1`.

Ações

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessReadyAsync\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)

- [GetTerminationTime\(\)](#)
- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)
- [Destroy\(\)](#)

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
Aws::GameLift::AwsStringOutcome Server::GetSdkVersion();
```

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto [the section called "AwsStringOutcome"](#). O objeto retornada inclui o número da versão (exemplo, 5.0.0). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =  
    Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma frota EC2 gerenciada. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método lê os parâmetros do servidor do ambiente host para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK();
```

Valor de retorno

Retorna um objeto [the section called “InitSDKOutcome”](#) que indica se o processo do servidor está pronto para ser chamado [ProcessReady\(\)](#).

Exemplo

```
//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
    Aws::GameLift::Server::InitSDK();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma Anywhere frota. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método requer parâmetros explícitos do servidor para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
Server::InitSDKOutcome Server::initSdkOutcome = InitSDK(serverParameters);
```

Parâmetros

[ServerParameters](#)

Para inicializar um servidor de jogo em uma GameLift Anywhere frota da Amazon, construa um `ServerParameters` objeto com as seguintes informações:

- O URL do WebSocket usado para se conectar ao seu servidor de jogo.
- O ID do processo usado para hospedar o servidor de jogos.
- O ID do computador que hospeda os processos do seu servidor de jogos.
- O ID da GameLift frota da Amazon que contém sua GameLift Anywhere computação da Amazon.

- O token de autorização gerado pela GameLift operação da Amazon.

Valor de retorno

Retorna um objeto [the section called "InitSDKOutcome"](#) que indica se o processo do servidor está pronto para ser chamado [ProcessReady\(\)](#).

Note

Se as chamadas para `InitSDK()` estiverem falhando para compilações de jogos implantadas em frotas do Anywhere, verifique o parâmetro `ServerSdkVersion` usado ao criar o recurso de compilação. Você deve definir explicitamente esse valor para a versão do SDK do servidor em uso. O valor padrão desse parâmetro é 4.x, o que não é compatível. Para resolver esse problema, crie uma nova versão e implante-a em uma nova frota.

Exemplo

GameLift AnywhereExemplo da Amazon

```
//Define the server parameters
std::string websocketUrl = "wss://us-west-1.api.amazongamelift.com";
std::string processId = "PID1234";
std::string fleetId = "arn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
std::string hostId = "HardwareAnywhere";
std::string authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
Aws::GameLift::Server::Model::ServerParameters serverParameters =
    Aws::GameLift::Server::Model::ServerParameters(webSocketUrl, authToken, fleetId,
    hostId, processId);

//Call InitSDK to establish a local connection with the GameLift agent to enable
    further communication.
Aws::GameLift::Server::InitSDKOutcome initSdkOutcome =
    Aws::GameLift::Server::InitSDK(serverParameters);
```

ProcessReady()

Notifica a Amazon de GameLift que o processo do servidor está pronto para hospedar sessões de jogo. Chame esse método após invocar [InitSDK\(\)](#). Esse método deve ser chamado somente uma vez por processo.

Sintaxe

```
GenericOutcome ProcessReady(const Aws::GameLift::Server::ProcessParameters
&processParameters);
```

Parâmetros

processParameters

Um objeto [ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes dos métodos de retorno de chamada implementados no código do servidor do jogo que o GameLift serviço da Amazon invoca para se comunicar com o processo do servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho para qualquer arquivo específico da sessão de jogo que você deseja que a Amazon capture e GameLift armazene.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra as implementações das funções de chamada e delegação [ProcessReady\(\)](#).

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");           // Example of a log file written by the
game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
  Aws::GameLift::Server::ProcessParameters(
    std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this),
    std::bind(&Server::OnHealthCheck, this),
    std::bind(&Server::OnUpdateGameSession, this),
    listenPort,
    Aws::GameLift::Server::LogParameters(logPaths)
```

```
);

Aws::GameLift::GenericOutcome outcome =
    Aws::GameLift::Server::ProcessReady(processReadyParameter);

// Implement callback functions
void Server::onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    GenericOutcome outcome =
        Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void Server::onProcessTerminate()
{
    // game-specific tasks required to gracefully shut down a game session,
    // such as notifying players, preserving game state data, and other cleanup
    GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool Server::onHealthCheck()
{
    bool health;
    // complete health evaluation within 60 seconds and set health
    return health;
}
```

ProcessReadyAsync()

Notifica o GameLift serviço da Amazon de que o processo do servidor está pronto para hospedar sessões de jogo. Este método deve ser chamado depois que o processo do servidor estiver pronto para hospedar uma sessão de jogo. Os parâmetros especificam os nomes das funções de retorno de chamada para GameLift a Amazon chamar em determinadas circunstâncias. O código do servidor de jogos deve implementar essas funções.

Essa chamada é assíncrona. Para fazer uma chamada síncrona, use [ProcessReady\(\)](#). Consulte [Inicializar o processo do servidor](#) para obter mais detalhes.

Sintaxe

```
GenericOutcomeCallable ProcessReadyAsync(
    const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parâmetros

processParameters

Um objeto [ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes dos métodos de retorno de chamada implementados no código do servidor do jogo que o GameLift serviço da Amazon invoca para se comunicar com o processo do servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho para qualquer arquivo específico da sessão de jogo que você deseja que a Amazon capture e GameLift armazene.

Exigido: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");           // This is an example of a log file
written by the game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);
int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
    Aws::GameLift::Server::ProcessParameters(std::bind(&Server::onStartGameSession, this,
std::placeholders::_1),
    std::bind(&Server::onProcessTerminate, this), std::bind(&Server::OnHealthCheck,
this),
    std::bind(&Server::OnUpdateGameSession, this), listenPort,
    Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcomeCallable outcome =
    Aws::GameLift::Server::ProcessReadyAsync(processReadyParameter);

// Implement callback functions
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
```

```
// game-specific tasks when starting a new game session, such as loading map
GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void onProcessTerminate()
{
    // game-specific tasks required to gracefully shut down a game session,
    // such as notifying players, preserving game state data, and other cleanup
    GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool onHealthCheck()
{
    // perform health evaluation and complete within 60 seconds
    return health;
}
```

ProcessEnding()

Notifica a Amazon de GameLift que o processo do servidor está sendo encerrado. Chame esse método depois de todas as outras tarefas de limpeza (incluindo o encerramento da sessão ativa do jogo) e antes de encerrar o processo. Dependendo do resultado `ProcessEnding()`, o processo sai com êxito (0) ou erro (-1) e gera um evento de frota. Se o processo for encerrado com um erro, o evento de frota gerado será `SERVER_PROCESS_TERMINATED_UNHEALTHY`.

Sintaxe

```
Aws::GameLift::GenericOutcome processEndingOutcome =
    Aws::GameLift::Server::ProcessEnding();
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo chama `ProcessEnding()` e `Destroy()` antes de encerrar o processo do servidor com um código de saída bem-sucedido ou de erro.

```
Aws::GameLift::GenericOutcome processEndingOutcome =
    Aws::GameLift::Server::ProcessEnding();
Aws::GameLift::Server::Destroy();
```

```
// Exit the process with success or failure
if (processEndingOutcome.IsSuccess()) {
    exit(0);
}
else {
    cout << "ProcessEnding() failed. Error: " <<
    processEndingOutcome.GetError().GetErrorMessage();
    exit(-1);
}
```

ActivateGameSession()

Notifica a Amazon de GameLift que o processo do servidor ativou uma sessão de jogo e agora está pronto para receber conexões de jogadores. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, após toda inicialização da sessão do jogo.

Sintaxe

```
Aws::GameLift::GenericOutcome activateGameSessionOutcome =
    Aws::GameLift::Server::ActivateGameSession();
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo mostra o chamado `ActivateGameSession()` como parte da função de delegação `onStartGameSession()`.

```
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession();
}
```

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador.

Sintaxe

```
GenericOutcome
UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCreationPolicy
newPlayerSessionPolicy);
```

Parâmetros

playerCreationSessionPolítica

Tipo: `PlayerSessionCreationPolicy` valor [enum](#).

Exigido: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
Aws::GameLift::GenericOutcome outcome =
    Aws::GameLift::Server::UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCr
```

GetGameSessionId()

Recupera o ID da sessão de jogo hospedada pelo processo do servidor ativo.

Para processos inativos que não são ativados com uma sessão de jogo, a chamada retorna um [the section called "GameLiftError"](#).

Sintaxe

```
AwsStringOutcome GetGameSessionId()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto [the section called "AwsStringOutcome"](#). Se não for bem-sucedido, retornará uma mensagem de erro.

Para processos inativos que não são ativados com uma sessão de jogo, a chamada retorna `Success=True` e `GameSessionId=""`.

Exemplo

```
Aws::GameLift::AwsStringOutcome sessionIdOutcome =  
    Aws::GameLift::Server::GetGameSessionId();
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado, se essa informação estiver disponível. Um processo do servidor entra em ação após receber um `onProcessTerminate()` retorno de chamada da Amazon GameLift. A Amazon GameLift liga `onProcessTerminate()` pelos seguintes motivos:

- Quando o processo do servidor relatou problemas de saúde ou não respondeu à Amazon GameLift.
- Ao encerrar a instância durante um evento de redução.
- Quando uma instância é encerrada devido a uma [interrupção na instância spot](#).

Sintaxe

```
AwsDateTimeOutcome GetTerminationTime()
```

Valor de retorno

Se for bem-sucedido, retornará o horário de término como um objeto `AwsDateTimeOutcome`. O valor é o tempo de rescisão, expresso em tiques decorridos desde então `0001 00:00:00`. Por exemplo, o valor da data e hora `2020-09-13 12:26:40 -000Z` é igual aos tiques `637355968000000000`. Se nenhum horário de encerramento estiver disponível, o retornará uma mensagem de erro.

Se o processo não tiver recebido um `ProcessParameters.OnProcessTerminate()` callback, uma mensagem de erro é retornada. Para obter mais informações sobre como desligar um processo do servidor, consulte [Responder a uma notificação de desligamento do processo do servidor](#).

Exemplo

```
Aws::GameLift::AwsLongOutcome TermTimeOutcome =  
    Aws::GameLift::Server::GetTerminationTime();
```

AcceptPlayerSession()

Notifica a Amazon de GameLift que um jogador com o ID de sessão de jogador especificado se conectou ao processo do servidor e precisa de validação. A Amazon GameLift verifica se o ID da sessão do jogador é válido. Depois que a sessão do jogador é validada, a Amazon GameLift altera o status do slot do jogador de RESERVADO para ATIVO.

Sintaxe

```
GenericOutcome AcceptPlayerSession(String playerId)
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo trata de uma solicitação de conexão que inclui a validação e a rejeição de IDs de sessão de jogadores inválidas.

```
void ReceiveConnectingPlayerSessionID (Connection& connection, const std::string&  
    playerId)  
{  
    Aws::GameLift::GenericOutcome connectOutcome =  
    Aws::GameLift::Server::AcceptPlayerSession(playerSessionId);  
    if(connectOutcome.IsSuccess())  
    {  
        connectionToSessionMap.emplace(connection, playerId);  
        connection.Accept();  
    }  
}
```



```
}
else
{
    connection.Reject(connectOutcome.GetError().GetMessage());
}
}
```

RemovePlayerSession()

Notifica a Amazon de GameLift que um jogador se desconectou do processo do servidor. Em resposta, a Amazon GameLift altera o slot do jogador para disponível.

Sintaxe

```
GenericOutcome RemovePlayerSession(String playerId)
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
Aws::GameLift::GenericOutcome disconnectOutcome =
    Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, que incluem configurações, metadados da sessão e dados do jogador. Use esse método para obter informações sobre o seguinte:

- Uma sessão para um jogador
- Todas as sessões de jogador em uma sessão de jogo
- Todas as sessões de jogadores associadas a um único ID de jogador

Sintaxe

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest  
describePlayerSessionsRequest)
```

Parâmetros

[DescribePlayerSessionsRequest](#)

Um objeto [the section called “DescribePlayerSessionsRequest”](#) que descreve quais sessões de jogador recuperar.

Valor de retorno

Se bem-sucedido, retorna um objeto [the section called “DescribePlayerSessionsOutcome”](#) que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação.

Exemplo

Este exemplo solicita todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Ao omitir NextToken e definir o valor limite como 10, a Amazon GameLift retorna os primeiros registros de sessão de 10 jogadores que correspondam à solicitação.

```
// Set request parameters  
Aws::GameLift::Server::Model::DescribePlayerSessionsRequest request;  
request.SetPlayerSessionStatusFilter(Aws::GameLift::Server::Model::PlayerSessionStatusMapper::G  
request.SetLimit(10);  
request.SetGameSessionId("the game session ID"); // can use GetGameSessionId()  
  
// Call DescribePlayerSessions  
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =  
    Aws::GameLift::Server::DescribePlayerSessions(request);
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Esta ação é assíncrona. Se novos jogadores forem combinados, a Amazon GameLift fornece dados atualizados do matchmaker usando a função de retorno de chamada. `OnUpdateGameSession()`

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest  
startBackfillRequest);
```

Parâmetros

[StartMatchBackfillRequest](#)

Um `StartMatchBackfillRequest` objeto que comunica as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essas informações são opcionais; se nenhum ID for fornecido, a Amazon GameLift gerará um.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor está nos dados do marcador de jogos da sessão de jogo.
- O ID da sessão de jogo a ser preenchida.
- Os dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Valor de retorno

Retorna um objeto [the section called "StartMatchBackfillOutcome"](#) com o ID do tíquete de alocação de correspondência ou um falha com uma mensagem de erro.

Exemplo

```
// Build a backfill request  
std::vector<Player> players;  
Aws::GameLift::Server::Model::StartMatchBackfillRequest startBackfillRequest;  
startBackfillRequest.SetTicketId("1111aaaa-22bb-33cc-44dd-5555eeee66ff"); // optional,  
    autogenerated if not provided  
startBackfillRequest.SetMatchmakingConfigurationArn("arn:aws:gamelift:us-  
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"); //from the game  
    session matchmaker data  
startBackfillRequest.SetGameSessionArn("the game session ARN"); // can use  
    GetGameSessionId()  
startBackfillRequest.SetPlayers(players); // from the  
    game session matchmaker data
```

```
// Send backfill request
Aws::GameLift::StartMatchBackfillOutcome backfillOutcome =
    Aws::GameLift::Server::StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void Server::OnUpdateGameSession(Aws::GameLift::Server::Model::GameSession gameSession,
    Aws::GameLift::Server::Model::UpdateReason updateReason, std::string backfillTicketId)
{
    // handle status messages
    // perform game-specific tasks to prep for newly matched players
}
```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Sintaxe

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parâmetros

[StopMatchBackfillRequest](#)

Um `StopMatchBackfillRequest` objeto identificando o tíquete de matchmaking a ser cancelado:

- O ID do ticket atribuído à solicitação de alocação.
- O marcador de jogo que recebeu a solicitação de alocação.
- A sessão do jogo associada à solicitação de alocação.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set backfill stop request parameters

Aws::GameLift::Server::Model::StopMatchBackfillRequest stopBackfillRequest;
```

```
stopBackfillRequest.SetTicketId("1111aaaa-22bb-33cc-44dd-5555eeee66ff");
stopBackfillRequest.SetGameSessionArn("the game session ARN"); // can use
  GetGameSessionId()
stopBackfillRequest.SetMatchmakingConfigurationArn("arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig");
// from the game session matchmaker data

Aws::GameLift::GenericOutcome stopBackfillOutcome =
  Aws::GameLift::Server::StopMatchBackfill(stopBackfillRequest);
```

GetComputeCertificate()

Recupera o caminho para o certificado TLS usado para criptografar a conexão de rede entre seu GameLift Anywhere recurso computacional da Amazon e a Amazon. GameLift Você pode usar o caminho do certificado ao registrar seu dispositivo computacional em uma GameLift Anywhere frota da Amazon. Para obter mais informações, consulte, [RegisterCompute](#).

Sintaxe

```
GetComputeCertificateOutcome Server::GetComputeCertificate()
```

Valor de retorno

Retorna um [the section called "GetComputeCertificateOutcome"](#).

Exemplo

```
Aws::GameLift::GetComputeCertificateOutcome certificate =
  Aws::GameLift::Server::GetComputeCertificate();
```

GetFleetRoleCredentials()

Recupera as credenciais da função do IAM que autorizam a GameLift Amazon a interagir com outras. Serviços da AWS Para ter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Sintaxe

```
GetFleetRoleCredentialsOutcome GetFleetRoleCredentials(GetFleetRoleCredentialsRequest
request);
```

Parâmetros

[GetFleetRoleCredentialsRequest](#)

Valor de retorno

Informa um objeto [the section called "GetFleetRoleCredentialsOutcome"](#).

Exemplo

```
// form the fleet credentials request
Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest
  getFleetRoleCredentialsRequest;
getFleetRoleCredentialsRequest.SetRoleArn("arn:aws:iam::123456789012:role/service-role/
exampleGameLiftAction");

Aws::GameLift::GetFleetRoleCredentialsOutcome credentials =
  Aws::GameLift::Server::GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Este exemplo mostra o uso do valor `RoleSessionName` opcional para atribuir um nome à sessão de credenciais para fins de auditoria. Se você não fornecer um nome de sessão de função, o valor padrão "*[fleet-id]-[host-id]*" será usado.

```
// form the fleet credentials request
Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest
  getFleetRoleCredentialsRequest;
getFleetRoleCredentialsRequest.SetRoleArn("arn:aws:iam::123456789012:role/service-role/
exampleGameLiftAction");
getFleetRoleCredentialsRequest.SetRoleSessionName("MyFleetRoleSession");

Aws::GameLift::GetFleetRoleCredentialsOutcome credentials =
  Aws::GameLift::Server::GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Destroy()

Libera o SDK do servidor de GameLift jogos da Amazon da memória. Como prática recomendada, chame esse método antes `ProcessEnding()` e depois de encerrar o processo. Se você estiver usando uma frota Anywhere e não estiver encerrando os processos do servidor após cada sessão de jogo, ligue `Destroy()` e reinicialize antes de notificar `InitSDK()` a Amazon de GameLift que o processo está pronto para hospedar uma sessão de jogo. `ProcessReady()`

Sintaxe

```
GenericOutcome Aws::GameLift::Server::Destroy();
```

Parâmetros

Não há parâmetros.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
Aws::GameLift::GenericOutcome processEndingOutcome =  
    Aws::GameLift::Server::ProcessEnding();  
Aws::GameLift::Server::Destroy();  
  
// Exit the process with success or failure  
if (processEndingOutcome.IsSuccess()) {  
    exit(0);  
}  
else {  
    cout << "ProcessEnding() failed. Error: " <<  
    processEndingOutcome.GetError().GetErrorMessage();  
    exit(-1);  
}
```

Referência GameLift do Amazon Server SDK (C++): tipos de dados

Você pode usar essa referência do SDK do servidor Amazon GameLift C++ para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon. GameLift Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Note

Este tópico descreve a API do Amazon GameLift C++ que você pode usar ao criar com a C++ Standard Library (`std`). Especificamente, essa documentação se aplica ao código que você compila com a opção `-DDGAMELIFT_USE_STD=1`.

Tipos de dados

- [LogParameters](#)
- [ProcessParameters](#)
- [UpdateGameSession](#)
- [GameSession](#)
- [ServerParameters](#)
- [StartMatchBackfillRequest](#)
- [Jogador](#)
- [DescribePlayerSessionsRequest](#)
- [StopMatchBackfillRequest](#)
- [AttributeValue](#)
- [GetFleetRoleCredentialsRequest](#)
- [AwsLongOutcome](#)
- [AwsStringOutcome](#)
- [DescribePlayerSessionsOutcome](#)
- [DescribePlayerSessionsResult](#)
- [GenericOutcome](#)
- [GenericOutcomeCallable](#)
- [PlayerSession](#)
- [StartMatchBackfillOutcome](#)
- [StartMatchBackfillResult](#)
- [GetComputeCertificateOutcome](#)
- [GetComputeCertificateResult](#)
- [GetFleetRoleCredentialsOutcome](#)
- [GetFleetRoleCredentialsResult](#)
- [InitSDKOutcome](#)
- [GameLiftError](#)
- [Enumerações](#)

LogParameters

Um objeto que identifica arquivos gerados durante uma sessão de jogo que você deseja que GameLift a Amazon envie e armazene após o término da sessão de jogo. O servidor do jogo fornece `LogParameters` à Amazon GameLift como parte de um `ProcessParameters` objeto em uma [ProcessReady\(\)](#) chamada.

Properties	Descrição
LogPaths	<p>A lista de caminhos de diretório para os arquivos de log do servidor de jogos que você deseja que GameLift a Amazon armazene para acesso futuro. O processo do servidor gera esses arquivos durante cada sessão de jogo. Você define caminhos e nomes de arquivos no servidor de jogo e os armazena no diretório raiz de compilação do jogo.</p> <p>Os caminhos de log devem ser absolutos. Por exemplo, se a compilação do jogo armazenar logs de sessão de jogo em um caminho como <code>MyGame\sessionLogs\</code>, o caminho estará <code>c:\game\MyGame\sessionLogs</code> em uma instância do Windows.</p> <p>Digite: <code>std::vector<std::string></code></p> <p>Obrigatório: não</p>

ProcessParameters

Esse tipo de dados contém o conjunto de parâmetros enviados para a Amazon GameLift em um [ProcessReady\(\)](#).

Properties	Descrição
LogParameters	Um objeto com caminhos de diretório para arquivos que são gerados durante uma sessão

	<p>de jogo. A Amazon GameLift copia e armazena os arquivos para acesso futuro.</p> <p>Digite: <code>Aws::GameLift::Server:: LogParameters</code></p> <p>Obrigatório: não</p>
OnHealthCheck	<p>A função de retorno de chamada que a Amazon GameLift invoca para solicitar um relatório de status de saúde do processo do servidor. A Amazon GameLift chama essa função a cada 60 segundos e espera 60 segundos por uma resposta. O processo do servidor retorna TRUE se estiver íntegro e FALSE se não estiver íntegro. Se nenhuma resposta for retornada, a Amazon GameLift registra o processo do servidor como não íntegro.</p> <p>Digite: <code>std::function<bool()> onHealthCheck</code></p> <p>Obrigatório: não</p>
OnProcessTerminate	<p>A função de retorno de chamada que a Amazon GameLift invoca para forçar o encerramento do processo do servidor. Depois de chamar essa função, a Amazon GameLift espera 5 minutos até que o processo do servidor seja encerrado e responda com uma ProcessEnding() chamada antes de encerrar o processo do servidor.</p> <p>Digite: <code>std::function<void()> onProcessTerminate</code></p> <p>Obrigatório: Sim</p>

OnRefreshConnection

O nome da função de retorno de chamada que a Amazon GameLift invoca para atualizar a conexão com o servidor do jogo.

Digite: `void OnRefreshConnectionDelegate()`

Obrigatório: Sim

OnStartGameSession

A função de retorno de chamada que a Amazon GameLift invoca para ativar uma nova sessão de jogo. A Amazon GameLift chama essa função em resposta a uma solicitação do cliente [CreateGameSession](#). A função de retorno de chamada passa um [GameSession](#) objeto, conforme definido na Amazon GameLift API Reference.

Digite: `const std::function<void (Aws::GameLift::Model::GameSession)> onStartGameSession`

Obrigatório: Sim

OnUpdateGameSession

A função de retorno de chamada que a Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. A Amazon GameLift chama essa função quando uma solicitação de preenchimento de correspondência é processada para fornecer dados atualizados do matchmaker. Ele passa um [GameSession](#) objeto, uma atualização de status (updateReason) e o ID do tíquete de preenchimento correspondente.

```
Digite: std::function<void(Aws::GameLift::Server::Model::UpdateGameSession)> onUpdateGameSession
```

Obrigatório: não

Porta

O número da porta em que o processo de servidor escuta para novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Digite: Integer

Obrigatório: Sim

UpdateGameSession

Esse tipo de dados é atualizado para um objeto de sessão de jogo, o que inclui o motivo pelo qual a sessão de jogo foi atualizada e o ID do tíquete de alocação relacionada, se a alocação for usada para preencher as sessões dos jogadores na sessão do jogo.

Propriedades	Descrição
GameSession	<p>Um GameSession objeto definido pela GameLift API da Amazon. O objeto <code>GameSession</code> contém propriedades que descrevem uma sessão de jogo.</p> <p>Digite: <code>Aws::GameLift::Server::GameSession</code></p> <p>Obrigatório: Sim</p>
UpdateReason	<p>O motivo pelo qual a sessão do jogo está sendo atualizada.</p> <p>Digite: <code>Aws::GameLift::Server::UpdateReason</code></p> <p>Obrigatório: Sim</p>
BackfillTicketId	<p>O ID do tíquete de alocação que está tentando atualizar a sessão do jogo.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

GameSession

Esse tipo de dados fornece detalhes de uma sessão de jogo.

Propriedades	Descrição
GameSessionId	<p>Um identificador exclusivo da sessão de jogo. Um ARN de sessão de jogo tem o seguinte formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> .</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
Nome	<p>Um rótulo descritivo da sessão do jogo.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
FleetId	<p>Um identificador exclusivo da frota em que a sessão de jogo está sendo executada.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
MaximumPlayerSessionCount	<p>O número máximo de conexões do jogador com a sessão do jogo.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>
Porta	<p>O número da porta para a sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta.</p> <p>Digite: <code>int</code></p>

Propriedades	Descrição
	Obrigatório: não
IpAddress	<p>O endereço IP da sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
GameSessionData	<p>Um conjunto de propriedades de sessão de jogo personalizadas, formatadas como um único valor de string.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
MatchmakerData	<p>Informações sobre o processo de marcação de jogos usado para criar a sessão do jogo, na sintaxe JSON, formatado como uma string. Além da configuração de marcação de jogos usada, ele contém dados sobre todos os jogadores atribuídos à partida, incluindo atributos do jogador e atribuições da equipe.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
GameProperties	<p>Um conjunto de propriedades personalizadas para uma sessão de jogo, formatado como pares de chave:valor. Essas propriedades são passadas com uma solicitação de início de uma nova sessão de jogo.</p> <p>Digite: <code>std :: vector < GameProperty ></code></p> <p>Obrigatório: não</p>
DnsName	<p>O identificador DNS atribuído à instância que está executando a sessão do jogo. Os valores têm o formato a seguir:</p> <ul style="list-style-type: none"> Frotas habilitadas para TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> Frotas não habilitadas para TLS: <code>ec2-<unique identifier>.compute.amazonaws.com</code> <p>Ao se conectar a uma sessão de jogo que está sendo executada em uma frota compatível com TLS, você deve usar o nome DNS, não o endereço IP.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

ServerParameters

Informações usadas para manter a conexão entre o servidor de jogos em uma GameLift Anywhere frota da Amazon e o GameLift serviço da Amazon. Essas informações são usadas ao iniciar novos

processos de servidor com [InitSDK\(\)](#). Para servidores hospedados em instâncias EC2 GameLift gerenciadas pela Amazon, use um objeto vazio.

Propriedades	Descrição
websocketUrl	<p>A <code>GameLiftServerSdkEndpoint</code> Amazon GameLift retorna quando você compra RegisterCompute um recurso GameLift Anywhere computacional da Amazon.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
processId	<p>Um identificador exclusivo registrado no processo do servidor que hospeda seu jogo.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
hostId	<p>O <code>HostID</code> é o <code>ComputeName</code> usado quando você registrou seu computador. Para obter mais informações, consulte, RegisterCompute.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
fleetId	<p>O identificador exclusivo da frota na qual o computador está registrado. Para obter mais informações, consulte, RegisterCompute.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
authToken	<p>O token de autenticação gerado pela Amazon GameLift que autentica seu servidor na</p>

Propriedades	Descrição
	<p>Amazon GameLift. Para obter mais informações, consulte, GetComputeAuthToken.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>

StartMatchBackfillRequest

Informações usadas para criar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações à Amazon GameLift em uma [StartMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	<p>Um identificador da sessão de jogo exclusivo . A operação de API retorna GetGameSessionId o identificador no formato de ARN.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
MatchmakingConfigurationArn	<p>Um identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. O marcador de jogos ARN da sessão de jogo original está no objeto de sessão de jogo na propriedade de dados do marcador de jogos. Saiba mais sobre os dados do marcador de jogos em Trabalhar com dados do marcador de jogos.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: Sim</p>
Jogadores	<p>Um conjunto de dados que representa todos os jogadores que estão na sessão de jogo.</p>

Propriedades	Descrição
	<p>O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais.</p> <p>Digite: <code>std::vector<Player></code></p> <p>Obrigatório: Sim</p>
TicketId	<p>Um identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se você não fornecer um valor, a Amazon GameLift gera um. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

Jogador

Esse tipo de dados representa um jogador em marcação de jogos. Ao iniciar uma solicitação de marcação de jogos, um jogador tem um ID de jogador, atributos, e possivelmente dados de latência. A Amazon GameLift adiciona informações da equipe após a realização de uma partida.

Propriedades	Descrição
LatencyInMS	<p>Um conjunto de valores expressos em milissegundos que indicam a quantidade de latência que um jogador experimenta quando conectado a um local.</p> <p>Se essa propriedade for usada, o jogador só será correspondido aos locais listados. Se um marcador de jogos tiver uma regra que avalie</p>

Propriedades	Descrição
	<p>a latência do jogador, os jogadores deverão reportar a latência a ser correspondida.</p> <p>Digite: <code>Dictionary<string, int></code></p> <p>Obrigatório: não</p>
PlayerAttributes	<p>Uma coleção de pares de chave:valor contendo informações de jogadores para uso em jogos de jogos. As chaves de atributos do jogador devem corresponder às PlayerAttributes usadas em um conjunto de regras de matchmaking.</p> <p>Para obter mais informações sobre os atributos do jogador, consulte AttributeValue.</p> <p>Digite: <code>std::map<std::string, AttributeValue></code></p> <p>Obrigatório: não</p>
PlayerId	<p>Um identificador exclusivo para um jogador.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
Team	<p>O nome da equipe à qual o jogador está designado em uma partida. Você define o nome da equipe no conjunto de regras de marcação de jogos.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

DescribePlayerSessionsRequest

Um objeto que especifica quais sessões de jogador devem ser recuperadas. O processo do servidor fornece essas informações com uma [DescribePlayerSessions\(\)](#) chamada para a Amazon GameLift.

Propriedades	Descrição
GameSessionId	<p>Um identificador da sessão de jogo exclusivo . Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada.</p> <p>O formato do ID da sessão do jogo é <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string></code>. A GameSessionID é uma string de ID personalizada ou uma</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
PlayerSessionId	<p>Um identificador exclusivo para uma sessão de jogador. Use esse parâmetro para solicitar uma única sessão de jogador específica.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
PlayerId	<p>Um identificador exclusivo para um jogador. Use este parâmetro para solicitar todas as sessões de jogador para um jogador específico. Consulte Gerar IDs de jogador.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
PlayerSessionStatusFilter	<p>O status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis incluem:</p> <ul style="list-style-type: none">• RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor nem foi validado.• ACTIVE – O player foi validado pelo processo do servidor e está conectado.• COMPLETED – A conexão do jogador foi interrompida.• TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou ou não foi validado dentro do tempo limite (60 segundos). <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
NextToken	<p>O token que indica o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
Limite	<p>O número máximo de resultados a serem retornados. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>

StopMatchBackfillRequest

Informações usadas para cancelar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações ao GameLift serviço da Amazon em uma [StopMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	<p>Um identificador exclusivo da sessão de jogo da solicitação que está sendo cancelada.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>
MatchmakingConfigurationArn	<p>Um identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>
TicketId	<p>Um identificador exclusivo do tíquete de solicitação de alocação a ser cancelado.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>

AttributeValue

Use esses valores em pares de chave-valor de atributo [Jogador](#). Esse objeto permite especificar um valor de atributo usando qualquer um dos tipos de dados válidos: string, número, matriz de string ou mapa de dados. Cada objeto `AttributeValue` deve usar exatamente uma das propriedades disponíveis: S, N, SL ou SDM.

Propriedades	Descrição
AttrType	<p>Especifica o tipo de valor do atributo. Os possíveis tipos de valor de atributo incluem:</p> <ul style="list-style-type: none"> • NONE • STRING • DOUBLE • STRING_LIST • STRING_DOUBLE_MAP <p>Obrigatório: não</p>
S	<p>Representa um valor de atributo de string.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
N	<p>Representa um valor de atributo numérico.</p> <p>Digite: <code>double</code></p> <p>Obrigatório: não</p>
SL	<p>Representa uma matriz de valores de atributos de string.</p> <p>Digite: <code>std::vector<std::string></code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
SDM	<p>Representa um dicionário de chaves de string e valores duplos.</p> <p>Digite: <code>std::map<std::string, double></code></p> <p>Obrigatório: não</p>

GetFleetRoleCredentialsRequest

Esse tipo de dados dá ao servidor do jogo acesso limitado aos seus outros recursos AWS. Para obter mais informações, consulte, [Configurar uma função de serviço do IAM para a Amazon GameLift](#).

Propriedades	Descrição
RoleArn	<p>O nome do recurso da Amazon (ARN) da perfil de serviço que estende o acesso limitado aos seus recursos AWS.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
RoleSessionName	<p>O nome da sessão da função que você pode usar para identificar uma AWS Security Token Service AssumeRole sessão de forma exclusiva. Esse nome é exposto em registros de auditoria, como os de CloudTrail.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

AwsLongOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: long Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto. Digite: long&& Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

AwsStringOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação.

Propriedades	Descrição
	Digite: <code>std::string</code> Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto. Digite: <code>long&&</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “GameLiftError” Obrigatório: não

DescribePlayerSessionsOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “DescribePlayerSessionsResult” Obrigatório: não

Propriedades	Descrição
ResultWithOwnership	<p>O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto.</p> <p>Digite: <code>Aws::GameLift::Server::Model::DescribePlayerSessionsResult</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

DescribePlayerSessionsResult

Uma coleção de objetos contendo propriedades para cada sessão de jogador que corresponde à solicitação.

Propriedades	Descrição
NextToken	<p>Um token que indica o início da próxima página sequencial de resultados. Use o token retornado com uma chamada anterior para essa operação. Para começar no início do conjunto de resultados, não especifique um valor. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.</p>

Propriedades	Descrição
	Digite: <code>std::string</code> Obrigatório: Sim
PlayerSessions	Digite: <code>IList<the section called "PlayerSession"></code> Obrigatório:
ResultWithOwnership	O resultado da ação, convertido como um <code>rvalue</code> , para que o código de chamada possa se apropriar do objeto. Digite: <code>std::string&&</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

GenericOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code>

Propriedades	Descrição
	Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

GenericOutcomeCallable

Esse tipo de dados é um resultado genérico assíncrono. Ele tem as seguintes propriedades:

Propriedades	Descrição
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

PlayerSession

Esse tipo de dados representa uma sessão de jogador que a Amazon GameLift passa para o servidor do jogo. Para obter mais informações, consulte [PlayerSession](#).

Propriedades	Descrição
CreationTime	Digite: long

Propriedades	Descrição
	Obrigatório: não
FleetId	Digite: <code>std::string</code> Obrigatório: não
GameSessionId	Digite: <code>std::string</code> Obrigatório: não
IpAddress	Digite: <code>std::string</code> Obrigatório: não
PlayerData	Digite: <code>std::string</code> Obrigatório: não
PlayerId	Digite: <code>std::string</code> Obrigatório: não
PlayerSessionId	Digite: <code>std::string</code> Obrigatório: não
Porta	Digite: <code>int</code> Obrigatório: não

Propriedades	Descrição
Status	<p>Status da sessão de jogador para filtrar resultados. Quando um <code>PlayerSessionId</code> ou <code>PlayerId</code> é fornecido, o <code>PlayerSessionStatusFilter</code> tem efeito na resposta.</p> <p>Tipo: Um <code>PlayerSessionStatus</code> enum. Os valores possíveis incluem o seguinte:</p> <ul style="list-style-type: none"> • ACTIVE • COMPLETED • NOT_SET • RESERVED • TIMEDOUT <p>Obrigatório: não</p>
TerminationTime	<p>Digite: <code>long</code></p> <p>Obrigatório: não</p>
DnsName	<p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

StartMatchBackfillOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	<p>O resultado da ação.</p> <p>Digite: the section called “StartMatchBackfillResult”</p>

Propriedades	Descrição
	Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto. Digite: <code>StartMatchBackfillResult&&</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

StartMatchBackfillResult

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
TicketId	Um identificador exclusivo para um tíquete de marcação de jogo. Se nenhum ID de bilhete for especificado aqui, a Amazon GameLift gerará um na forma de um UUID. Use esse identificador para rastrear o status do ticket de alocação da partida e recuperar os resultados da partida. Digite: <code>std::string</code>

Propriedades	Descrição
	Obrigatório: não

GetComputeCertificateOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	<p>O resultado da ação.</p> <p>Digite: the section called “GetComputeCertificateResult”</p> <p>Obrigatório: não</p>
ResultWithOwnership	<p>O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto.</p> <p>Digite: <code>Aws::GameLift::Server::Model::GetComputeCertificateResult&&</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

GetComputeCertificateResult

O caminho para o certificado TLS em seu computador e o nome do host do computador.

Propriedades	Descrição
CertificatePath	<p>O caminho para o certificado TLS em seu recurso computacional. Ao usar uma frota GameLift gerenciada pela Amazon, esse caminho contém:</p> <ul style="list-style-type: none"> • <code>certificate.pem</code> : o certificado do usuário final. A cadeia completa de certificados é a combinação de <code>certificateChain.pem</code> anexados a esse certificado. • <code>certificateChain.pem</code> : a cadeia de certificados que contém o certificado raiz e os certificados intermediários. • <code>rootCertificate.pem</code> : Certificado raiz. • <code>privateKey.pem</code> : a chave privada para o certificado do usuário final. <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
ComputeName	<p>O nome do seu recurso computacional.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

GetFleetRoleCredentialsOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	<p>O resultado da ação.</p> <p>Digite: the section called “GetFleetRoleCredentialsResult”</p> <p>Obrigatório: não</p>
ResultWithOwnership	<p>O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto.</p> <p>Digite: <code>Aws::GameLift::Server::Model::GetFleetRoleCredentialsResult</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

GetFleetRoleCredentialsResult

Propriedades	Descrição
AccessKeyId	<p>O ID da chave de acesso para autenticar e fornecer acesso aos seus recursos AWS.</p> <p>Digite: <code>string</code></p>

Propriedades	Descrição
	Obrigatório: não
AssumedRoleId	O ID do usuário ao qual o perfil de serviço pertence. Digite: <code>string</code> Obrigatório: não
AssumedRoleUserArn	O nome do recurso da Amazon (ARN) do usuário ao perfil de serviço de serviço pertence. Digite: <code>string</code> Obrigatório: não
Expiração	A quantidade de tempo até que suas credenciais de sessão expirem. Digite: <code>DateTime</code> Obrigatório: não
SecretAccessKey	O ID da chave de acesso secreta para autenticação. Digite: <code>string</code> Obrigatório: não
SessionToken	Um token para identificar a sessão ativa atual interagindo com seus recursos AWS. Digite: <code>string</code> Obrigatório: não

Propriedades	Descrição
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: bool</p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

InitSDKOutcome

Note

`InitSDKOutcome` é retornado somente quando você cria o SDK com a bandeira `std`. Se você construir com a `nostd` bandeira, ela será [the section called “GenericOutcome”](#) retornada em vez disso.

Propriedades	Descrição
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: bool</p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

GameLiftError

Propriedades	Descrição
ErrorType	O tipo de erro. Tipo: Um <code>GameLiftErrorType</code> enum . Obrigatório: não
ErrorMessage	A mensagem de erro. Digite: <code>std::string</code> Obrigatório: não
ErrorMessage	A mensagem de erro. Digite: <code>std::string</code> Obrigatório: não
ErrorMessage	A mensagem de erro. Digite: <code>std::string</code> Obrigatório: não

Enumerações

Os enums definidos para o Amazon GameLift Server SDK (C++) são definidos da seguinte forma:

GameLiftErrorType

Valor da string indicando o tipo de erro. Os valores válidos são:

- `BAD_REQUEST_EXCEPTION`
- `GAMESESSION_ID_NOT_SET` – O ID da sessão do jogo não foi definido.
- `INTERNAL_SERVICE_EXCEPTION`
- `LOCAL_CONNECTION_FAILED` — A conexão local com a Amazon falhou. GameLift
- `NETWORK_NOT_INITIALIZED` – A rede não foi inicializada.
- `SERVICE_CALL_FAILED` – Uma chamada AWS para um serviço falhou.
- `WEBSOCKET_CONNECT_FAILURE`
- `WEBSOCKET_CONNECT_FAILURE_FORBIDDEN`
- `WEBSOCKET_CONNECT_FAILURE_INVALID_URL`

- `WEBSOCKET_CONNECT_FAILURE_TIMEOUT`
- `ALREADY_INITIALIZED` — O servidor ou cliente GameLift da Amazon já foi inicializado com `Initialize ()`.
- `FLEET_MISMATCH` – A frota alvo não corresponde à frota de uma `GameSession` ou `PlayerSession`.
- `GAMELIFT_CLIENT_NOT_INITIALIZED` — O cliente Amazon não foi inicializado. GameLift
- `GAMELIFT_SERVER_NOT_INITIALIZED` — O servidor Amazon não foi inicializado. GameLift
- `GAME_SESSION_ENDED_FAILED` — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que a sessão do jogo foi encerrada.
- `GAME_SESSION_NOT_READY` — A sessão de jogo do GameLift Amazon Server não foi ativada.
- `GAME_SESSION_READY_FAILED` — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que a sessão do jogo está pronta.
- `INITIALIZATION_MISMATCH` – Um método de cliente foi chamado depois de `Server::Initialize()` ou vice-versa.
- `NOT_INITIALIZED` — O GameLift servidor ou cliente da Amazon não foi inicializado com `Initialize ()`.
- `NO_TARGET_ALIASID_SET` – Um `aliasID` de destino não foi definido.
- `NO_TARGET_FLEET_SET` – Uma frota de destino não foi definida.
- `PROCESS_ENDING_FAILED` — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que o processo está sendo encerrado.
- `PROCESS_NOT_ACTIVE` — O processo do servidor ainda não está ativo, não está vinculado a um e não pode `GameSession` aceitar ou processar. `PlayerSessions`
- `PROCESS_NOT_READY` – O processo do servidor ainda não está pronto para ser ativado.
- `PROCESS_READY_FAILED` — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que o processo está pronto.
- `SDK_VERSION_DETECTION_FAILED` – Falha na detecção da versão do SDK.
- `STX_CALL_FAILED` – Uma chamada para o componente de back-end do servidor xSTx falhou.
- `STX_INITIALIZATION_FAILED` – O componente de back-end do servidor xSTx falhou ao inicializar.
- `UNEXPECTED_PLAYER_SESSION` – Uma sessão de jogador não registrada foi encontrada pelo servidor.

- WEBSOCKET_CONNECT_FAILURE
- WEBSOCKET_CONNECT_FAILURE_FORBIDDEN
- WEBSOCKET_CONNECT_FAILURE_INVALID_URL
- WEBSOCKET_CONNECT_FAILURE_TIMEOUT
- WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE — Falha recuperável ao enviar uma mensagem ao Serviço. GameLift WebSocket
- WEBSOCKET_SEND_MESSAGE_FAILURE — Falha ao enviar uma mensagem para o serviço. GameLift WebSocket
- MATCH_BACKFILL_REQUEST_VALIDATION – A validação da solicitação falhou.
- PLAYER_SESSION_REQUEST_VALIDATION – A validação da solicitação falhou.

PlayerSessionCreationPolicy

Valor de string que indica se a sessão do jogo aceita novos jogadores. Os valores válidos são:

- ACCEPT_ALL – Aceite todas as novas sessões de jogador.
- DENY_ALL – Recuse todas as novas sessões de jogador.
- NOT_SET – A sessão do jogo não está configurada para aceitar ou negar sessões de novos jogadores.

Referência do SDK 3.x do servidor C++ do Amazon GameLift

Você poderá usar essa referência do SDK 3.x do servidor C++ do Amazon GameLift para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência do SDK do servidor Amazon GameLift \(C++\): ações](#)
- [Referência do SDK do servidor Amazon GameLift \(C++\): tipos de dados](#)

Referência do SDK do servidor Amazon GameLift (C++): ações

Você poderá usar essa referência do SDK do servidor do Amazon GameLift C++ para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Ações

- [AcceptPlayerSession\(\)](#)
- [ActivateGameSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetInstanceCertificate\(\)](#)
- [GetSdkVersion\(\)](#)
- [GetTerminationTime\(\)](#)
- [InitSDK\(\)](#)
- [ProcessEnding\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessReadyAsync\(\)](#)
- [RemovePlayerSession\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [TerminateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [Destroy\(\)](#)

AcceptPlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se conectou ao processo do servidor e precisa de validação. O Amazon GameLift verifica se o ID da sessão do jogador é válido, ou seja, se o ID do jogador reservou um slot de jogador na sessão do jogo. Depois de validado, o Amazon GameLift altera o status do slot do jogador de RESERVED para ACTIVE.

Sintaxe

```
GenericOutcome AcceptPlayerSession(const std::string& playerSessionId);
```

Parâmetros

playerSessionId

ID exclusiva emitida pelo serviço do Amazon GameLift em resposta a uma chamada para a ação [CreatePlayerSession](#) da API do AWSSDK do Amazon GameLift. O cliente do jogo faz referência a esse ID ao se conectar ao processo do servidor.

Tipo: `std::string`

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra uma função para processar uma solicitação de conexão, inclusive validando e rejeitando IDs de sessão do jogador inválidos.

```
void ReceiveConnectingPlayerSessionID (Connection& connection, const std::string&
playerSessionId){
    Aws::GameLift::GenericOutcome connectOutcome =
        Aws::GameLift::Server::AcceptPlayerSession(playerSessionId);
    if(connectOutcome.IsSuccess())
    {
        connectionToSessionMap.emplace(connection, playerSessionId);
        connection.Accept();
    }
    else
    {
        connection.Reject(connectOutcome.GetError().GetMessage());
    }
}
```

ActivateGameSession()

Notifica o serviço do Amazon GameLift de que o processo do servidor iniciou uma sessão de jogo e está pronto para receber as conexões dos jogadores. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, depois que toda a inicialização da sessão tiver sido concluída.

Sintaxe

```
GenericOutcome ActivateGameSession();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo mostra `ActivateGameSession()` sendo chamado como parte da função de retorno de chamada `onStartGameSession()`.

```
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession();
}
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, inclusive configurações, metadados da sessão e dados do jogador. Use essa ação para obter informações de uma única sessão de jogador, para todas as sessões de jogador em uma sessão de jogo, ou para todas as sessões de jogador associadas a um único ID de jogador.

Sintaxe

```
DescribePlayerSessionsOutcome DescribePlayerSessions (
    const Aws::GameLift::Server::Model::DescribePlayerSessionsRequest
    &describePlayerSessionsRequest);
```

Parâmetros

describePlayerSessionsRequest

Um objeto [DescribePlayerSessionsRequest](#) que descreve quais sessões de jogador recuperar.

Obrigatório: Sim

Valor de retorno

Se bem-sucedido, retorna um objeto `DescribePlayerSessionsOutcome` que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação. Os objetos de sessão de jogador têm uma estrutura idêntica à do tipo de dados [PlayerSession](#) da API do SDKAWS do Amazon GameLift.

Exemplo

Este exemplo ilustra uma solicitação para todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Omitindo `NextToken` e definindo o valor `Limit` como 10, o Amazon GameLift retorna os 10 primeiros registros de sessões do jogador correspondentes à solicitação.

```
// Set request parameters
Aws::GameLift::Server::Model::DescribePlayerSessionsRequest request;
request.SetPlayerSessionStatusFilter(Aws::GameLift::Server::Model::PlayerSessionStatusMapper::G
request.SetLimit(10);
request.SetGameSessionId("the game session ID");    // can use GetGameSessionId()

// Call DescribePlayerSessions
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =
    Aws::GameLift::Server::DescribePlayerSessions(request);
```

GetGameSessionId()

Recupera um identificador exclusivo para a sessão do jogo hospedada no momento pelo processo do servidor, caso o processo do servidor esteja ativo. O identificador é retornado no formato de ARN: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`.

Para processos inativos que ainda não foram ativados com uma sessão de jogo, a chamada retorna `Success=True` e `GameSessionId=""` (uma string vazia).

Sintaxe

```
AwsStringOutcome GetGameSessionId();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto `AwsStringOutcome`. Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsStringOutcome sessionIdOutcome =  
    Aws::GameLift::Server::GetGameSessionId();
```

GetInstanceCertificate()

Recupera a localização do arquivo de um certificado TLS codificado por pem que está associado à frota e suas instâncias. O AWS Certificate Manager gera esse certificado quando você cria uma nova frota com a configuração do certificado definida como `GENERATED`. Use esse certificado para estabelecer uma conexão segura com um cliente de jogo e para criptografar a comunicação entre cliente e servidor.

Sintaxe

```
GetInstanceCertificateOutcome GetInstanceCertificate();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se for bem-sucedido, o retorno será um objeto `GetInstanceCertificateOutcome` que contém a localização do arquivo de certificado TLS e da cadeia de certificados da frota, que estão armazenados na instância. Um arquivo de certificado raiz, extraído da cadeia de certificados, também é armazenado na instância. Se não for bem-sucedido, retornará uma mensagem de erro.

Para obter mais informações sobre o certificado e os dados da cadeia de certificados, consulte [Elementos de resposta GetCertificate](#) na Referência da API do AWS Certificate Manager.

Exemplo

```
Aws::GameLift::GetInstanceCertificateOutcome certificateOutcome =  
    Aws::GameLift::Server::GetInstanceCertificate();
```

GetSdkVersion()

Retorna o número da versão atual do SDK em uso.

Sintaxe

```
AwsStringOutcome GetSdkVersion();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto `AwsStringOutcome`. A string retornada inclui apenas o número da versão (exemplo, "3.1.5"). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =  
    Aws::GameLift::Server::GetSdkVersion();
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado, se essa informação estiver disponível. Um processo do servidor realiza essa ação depois de receber um retorno de chamada `onProcessTerminate()` do serviço do Amazon GameLift. O Amazon GameLift pode chamar `onProcessTerminate()` pelos seguintes motivos: (1) quando o processo do servidor relatou problemas de saúde ou não respondeu ao Amazon GameLift; (2) ao encerrar a instância durante um evento de redução ou (3) quando uma instância está sendo encerrada devido a uma [interrupção do Spot](#).

Se o processo tiver recebido um retorno de chamada `onProcessTerminate()`, o valor retornado será o tempo estimado de encerramento. Se o processo não tiver recebido um retorno de chamada `onProcessTerminate()`, uma mensagem de erro será retornada. Saiba mais sobre [como desligar um processo do servidor](#).

Sintaxe

```
AwsLongOutcome GetTerminationTime();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se for bem-sucedido, retornará o horário de término como um objeto `AwsLongOutcome`. O valor é o tempo de término, expresso em tiques decorridos desde 0001 00:00:00. Por exemplo, o valor da data e hora 2020-09-13 12:26:40 -000Z é igual aos tiques 637355968000000000. Se nenhum horário de rescisão estiver disponível, o retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsLongOutcome TermTimeOutcome =  
    Aws::GameLift::Server::GetTerminationTime();
```

InitSDK()

Inicializa o SDK do Amazon GameLift. Esse método deve ser chamado na inicialização antes de qualquer outra inicialização relacionada ao Amazon GameLift.

Sintaxe

```
InitSDKOutcome InitSDK();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará um objeto `InitSdkOutcome` indicando que o processo de servidor está pronto para chamar [ProcessReady\(\)](#).

Exemplo

```
Aws::GameLift::Server::InitSDKOutcome initOutcome =  
    Aws::GameLift::Server::InitSDK();
```

ProcessEnding()

Notifica o serviço do Amazon GameLift de que o processo do servidor está sendo desligado. Esse método deverá ser chamado depois de todas as outras tarefas de limpeza, inclusive desligar todas

as sessões de jogos ativas. Este método deve sair com um código de saída zero; um código de saída diferente de zero resulta em uma mensagem de evento indicando que o processo não foi encerrado corretamente.

Depois que o método sair com um código de 0, você poderá encerrar o processo com um código de saída bem-sucedido. Você também pode rásair do processo com um código de erro. Se você sair com um código de erro, o evento da frota indicará que o processo foi encerrado de forma anormal (SERVER_PROCESS_TERMINATED_UNHEALTHY).

Sintaxe

```
GenericOutcome ProcessEnding();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
Aws::GameLift::GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
if (outcome.Success)
    exit(0); // exit with success
// otherwise, exit with error code
exit(errorCode);
```

ProcessReady()

Notifica o serviço do Amazon GameLift de que o processo do servidor está pronto para hospedar sessões do jogo. Chame esse método depois de invocar [InitSDK\(\)](#) e concluir todas as tarefas de configuração necessárias antes que o processo do servidor possa hospedar uma sessão do jogo. Esse método deve ser chamado somente uma vez por processo.

Essa chamada é síncrona. Para fazer uma chamada assíncrona, use [ProcessReadyAsync\(\)](#). Consulte [Inicializar o processo do servidor](#) para obter mais detalhes.

Sintaxe

```
GenericOutcome ProcessReady(  
    const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parâmetros

processParameters

Um objeto [ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes de métodos de retorno de chamada, implementados no código do servidor de jogos, que o serviço do Amazon GameLift invoca para se comunicar com o processo de servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho de qualquer arquivo específico da sessão do jogo que você deseja que o Amazon GameLift capture e armazene.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra as implementações das funções de chamada e retorno de chamada [ProcessReady\(\)](#).

```
// Set parameters and call ProcessReady  
std::string serverLog("serverOut.log");           // Example of a log file written by the  
game server  
std::vector<std::string> logPaths;  
logPaths.push_back(serverLog);  
  
int listenPort = 9339;  
  
Aws::GameLift::Server::ProcessParameters processReadyParameter =  
    Aws::GameLift::Server::ProcessParameters(  
        std::bind(&Server::onStartGameSession, this, std::placeholders::_1),  
        std::bind(&Server::onProcessTerminate, this),  
        std::bind(&Server::OnHealthCheck, this),
```

```
std::bind(&Server::OnUpdateGameSession, this),
listenPort,
Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcome outcome =
    Aws::GameLift::Server::ProcessReady(processReadyParameter);

// Implement callback functions
void Server::onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    GenericOutcome outcome =
        Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void Server::onProcessTerminate()
{
    // game-specific tasks required to gracefully shut down a game session,
    // such as notifying players, preserving game state data, and other cleanup
    GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool Server::onHealthCheck()
{
    bool health;
    // complete health evaluation within 60 seconds and set health
    return health;
}
```

ProcessReadyAsync()

Notifica o serviço do Amazon GameLift de que o processo do servidor está pronto para hospedar sessões do jogo. Este método deve ser chamado assim que o processo do servidor estiver pronto para hospedar uma sessão de jogo. Os parâmetros especificam os nomes das funções de retorno do Amazon GameLift para chamar determinadas circunstâncias. O código do servidor de jogos deve implementar essas funções.

Essa chamada é assíncrona. Para fazer uma chamada síncrona, use [ProcessReady\(\)](#). Consulte [Inicializar o processo do servidor](#) para obter mais detalhes.

Sintaxe

```
GenericOutcomeCallable ProcessReadyAsync(
```

```
const Aws::GameLift::Server::ProcessParameters &processParameters);
```

Parâmetros

processParameters

Um objeto [ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes de métodos de retorno de chamada, implementados no código do servidor de jogos, que o serviço do Amazon GameLift invoca para se comunicar com o processo de servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho de qualquer arquivo específico da sessão do jogo que você deseja que o Amazon GameLift capture e armazene.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set parameters and call ProcessReady
std::string serverLog("serverOut.log");           // This is an example of a log file
written by the game server
std::vector<std::string> logPaths;
logPaths.push_back(serverLog);

int listenPort = 9339;

Aws::GameLift::Server::ProcessParameters processReadyParameter =
    Aws::GameLift::Server::ProcessParameters(
        std::bind(&Server::onStartGameSession, this, std::placeholders::_1),
        std::bind(&Server::onProcessTerminate, this),
        std::bind(&Server::OnHealthCheck, this),
        std::bind(&Server::OnUpdateGameSession, this),
        listenPort,
        Aws::GameLift::Server::LogParameters(logPaths));

Aws::GameLift::GenericOutcomeCallable outcome =
    Aws::GameLift::Server::ProcessReadyAsync(processReadyParameter);
```

```
// Implement callback functions
void onStartGameSession(Aws::GameLift::Model::GameSession myGameSession)
{
    // game-specific tasks when starting a new game session, such as loading map
    GenericOutcome outcome = Aws::GameLift::Server::ActivateGameSession (maxPlayers);
}

void onProcessTerminate()
{
    // game-specific tasks required to gracefully shut down a game session,
    // such as notifying players, preserving game state data, and other cleanup
    GenericOutcome outcome = Aws::GameLift::Server::ProcessEnding();
}

bool onHealthCheck()
{
    // perform health evaluation and complete within 60 seconds
    return health;
}
```

RemovePlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se desconectou do processo do servidor. Em resposta, o Amazon GameLift altera o slot do jogador para um disponível, o que permite ser atribuído a um novo jogador.

Sintaxe

```
GenericOutcome RemovePlayerSession(
    const std::string& playerSessionId);
```

Parâmetros

playerSessionId

ID exclusiva emitida pelo serviço do Amazon GameLift em resposta a uma chamada para a ação [CreatePlayerSession](#) da API do AWS SDK Amazon GameLift. O cliente do jogo faz referência a esse ID ao se conectar ao processo do servidor.

Tipo: `std::string`

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
Aws::GameLift::GenericOutcome disconnectOutcome =  
    Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Consulte também a ação [StartMatchBackfill\(\)](#) do SDK da AWS. Com essa ação, as solicitações de alocação de correspondência podem ser iniciadas por um processo do servidor de jogos que esteja hospedando a sessão do jogo. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Esta ação é assíncrona. Se a correspondência dos novos jogadores for bem-sucedida, o serviço do Amazon GameLift entregará os dados atualizados do marcador de jogos invocando a função de retorno de chamada `OnUpdateGameSession()`.

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
StartMatchBackfillOutcome StartMatchBackfill (  
    const Aws::GameLift::Server::Model::StartMatchBackfillRequest  
    &startBackfillRequest);
```

Parâmetros

StartMatchBackfillRequest

Um objeto [StartMatchBackfillRequest](#) que fornece as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essa informação é opcional. Caso nenhum ID seja fornecido, o Amazon GameLift gerará um automaticamente.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor pode ser obtido dos dados do marcador da sessão do jogo.

- ID da sessão de jogo que está sendo alocada.
- Dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Obrigatório: Sim

Valor de retorno

Retorna um objeto `StartMatchBackfillOutcome` com o tíquete de alocação da correspondência ou um falha com uma mensagem de erro. O status do tíquete pode ser acompanhado usando a ação do SDK da AWS [DescribeMatchmaking\(\)](#).

Exemplo

```
// Build a backfill request
std::vector<Player> players;
Aws::GameLift::Server::Model::StartMatchBackfillRequest startBackfillRequest;
startBackfillRequest.SetTicketId("a ticket ID");
    //optional, autogenerated if not provided
startBackfillRequest.SetMatchmakingConfigurationArn("the matchmaker configuration
ARN"); //from the game session matchmaker data
startBackfillRequest.SetGameSessionArn("the game session ARN");
    // can use GetGameSessionId()
startBackfillRequest.SetPlayers(players);
    //from the game session matchmaker data

// Send backfill request
Aws::GameLift::StartMatchBackfillOutcome backfillOutcome =
    Aws::GameLift::Server::StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void Server::OnUpdateGameSession(Aws::GameLift::Server::Model::GameSession gameSession,
    Aws::GameLift::Server::Model::UpdateReason updateReason, std::string backfillTicketId)
{
    // handle status messages
    // perform game-specific tasks to prep for newly matched players
}
```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa criada com [StartMatchBackfill\(\)](#). Consulte também a ação [StopMatchmaking\(\)](#) do SDK da AWS. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Sintaxe

```
GenericOutcome StopMatchBackfill (  
    const Aws::GameLift::Server::Model::StopMatchBackfillRequest &stopBackfillRequest);
```

Parâmetros

StopMatchBackfillRequest

Um objeto [StopMatchBackfillRequest](#) que identifica o tíquete de marcação de jogos para cancelar:

- ID do tíquete atribuído à solicitação de alocação sendo cancelada
- o marcador de jogo que recebeu a solicitação de alocação
- a sessão do jogo associada à solicitação de alocação

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set backfill stop request parameters  
  
Aws::GameLift::Server::Model::StopMatchBackfillRequest stopBackfillRequest;  
stopBackfillRequest.SetTicketId("the ticket ID");  
stopBackfillRequest.SetGameSessionArn("the game session ARN");  
    // can use GetGameSessionId()  
stopBackfillRequest.SetMatchmakingConfigurationArn("the matchmaker configuration ARN");  
    // from the game session matchmaker data  
  
Aws::GameLift::GenericOutcome stopBackfillOutcome =  
    Aws::GameLift::Server::StopMatchBackfillRequest(stopBackfillRequest);
```

TerminateGameSession()

Esse método foi descontinuado com a versão 4.0.1. Em vez disso, o processo do servidor deve ser chamado [ProcessEnding\(\)](#) após o término de uma sessão de jogo.

Notifica o serviço do Amazon GameLift de que o processo do servidor encerrou a sessão atual do jogo. Essa ação é chamada quando o processo do servidor permanece ativo e pronto para hospedar uma nova sessão de jogo. Ele deve ser chamado somente após a conclusão do procedimento de encerramento da sessão de jogo, pois indica ao Amazon GameLift que o processo do servidor está imediatamente disponível para hospedar uma nova sessão de jogo.

Essa ação não será chamada se o processo do servidor for encerrado após o término da sessão do jogo. Em vez disso, chame [ProcessEnding\(\)](#) para sinalizar que tanto a sessão do jogo quanto o processo do servidor estão terminando.

Sintaxe

```
GenericOutcome TerminateGameSession();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Consulte também a ação do SDK da AWS [UpdateGameSession\(\)](#).

Sintaxe

```
GenericOutcome UpdatePlayerSessionCreationPolicy(  
    Aws::GameLift::Model::PlayerSessionCreationPolicy newPlayerSessionPolicy);
```

Parâmetros

newPlayerSessionPolicy

newPlayerSessionPolicy

Tipo: enumeração `Aws::GameLift::Model::PlayerSessionCreationPolicy`. Os valores válidos são:

- `ACCEPT_ALL` – Aceite todas as novas sessões de jogador.

- DENY_ALL – Recuse todas as novas sessões de jogador.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
Aws::GameLift::GenericOutcome outcome =  
    Aws::GameLift::Server::UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::PlayerSessionCr
```

Destroy()

Limpa a memória alocada por `initSDK ()` durante a inicialização do servidor do jogo. Use esse método depois de finalizar um processo no servidor do jogo para evitar o desperdício de memória do servidor.

Sintaxe

```
GenericOutcome Aws::GameLift::Server::Destroy();
```

Parâmetros

Não há parâmetros.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo limpa a memória alocada pelo `initSDK` após o término de um processo no servidor de jogos.

```
if (Aws::GameLift::Server::ProcessEnding().IsSuccess()) {  
    Aws::GameLift::Server::Destroy();  
    exit(0);  
}
```

```
}
```

Referência do SDK do servidor Amazon GameLift (C++): tipos de dados

Você poderá usar essa referência do SDK de servidor do Amazon GameLift C++ para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Esta API é definida em `GameLiftServerAPI.h`, `LogParameters.h` e `ProcessParameters.h`.

- [Ações](#)
- Tipos de dados

DescribePlayerSessionsRequest

Esse tipo de dados é usado para especificar quais sessões de jogador recuperar. Você pode usá-lo da seguinte forma:

- Forneça um `PlayerSessionId` para solicitar uma sessão de jogador específica.
- Forneça um `GameSessionId` para solicitar todas as sessões de jogador na sessão do jogo especificada.
- Forneça um `PlayerId` para solicitar todas as sessões de jogador para o jogador especificado.

Para grandes coleções de sessões de jogador, use os parâmetros de paginação para recuperar resultados em blocos sequenciais.

Índice

GameSessionId

Identificador da sessão de jogo exclusivo. Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada. O formato do ID da sessão de jogo é o seguinte: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. O valor de `<ID string>` é uma string de ID personalizada ou (se houver uma especificada quando a sessão de jogo foi criada) uma string gerada.

Tipo: sequência

Obrigatório: não

Limite

O número máximo de resultados a serem retornados. Use esse parâmetro com NextToken para obter resultados como um conjunto de páginas sequenciais. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: inteiro

Obrigatório: não

NextToken

Token indicando o início da próxima página sequencial de resultados. Use o token retornado com uma chamada anterior para essa ação. Para especificar o início do conjunto de resultados, não especifique um valor. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: sequência

Obrigatório: não

PlayerId

Identificador exclusivo para um jogador. Os IDs de jogador são definidos pelo desenvolvedor. Consulte [Gerar IDs de jogador](#).

Tipo: sequência

Obrigatório: não

PlayerSessionId

Identificador exclusivo para uma sessão de jogador.

Tipo: sequência

Obrigatório: não

PlayerSessionStatusFilter

Status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis estão os seguintes:

- RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor e/ou foi validado.
- ACTIVE – O jogador foi validado pelo processo de servidor e está conectado no momento.

- COMPLETED – A conexão do jogador foi interrompida.
- TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou e/ou não foi validado dentro do tempo limite (60 segundos).

Tipo: sequência

Obrigatório: não

LogParameters

Este tipo de dados é usado para identificar quais arquivos gerados durante uma sessão de jogo você deseja que o Amazon GameLift faça upload e armazene assim que a sessão de jogo terminar. Essas informações são comunicadas ao serviço do Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Índice

logPaths

Caminhos de diretório para arquivos de log do servidor de jogos que você deseja que o Amazon GameLift armazene para acesso futuro. Esses arquivos são gerados durante cada sessão do jogo. Os nomes e caminhos de arquivos são definidos no seu servidor de jogos e armazenados no diretório raiz da compilação do jogo. Os caminhos de log devem ser caminhos absolutos. Por exemplo, se a compilação do jogo armazenasse logs de sessão de jogo em um caminho como `MyGame\sessionlogs\`, o caminho de log seria `c:\game\MyGame\sessionLogs` (em uma instância do Windows) ou `/local/game/MyGame/sessionLogs` (em uma instância do Linux).

Tipo: `std::vector<std::string>`

Obrigatório: não

ProcessParameters

Este tipo de dados contém o conjunto de parâmetros enviado para o serviço do Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Índice

port

Número da porta em que o processo de servidor escuta para novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa

compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Tipo: inteiro

Obrigatório: Sim

logParameters

Objeto com uma lista de caminhos de diretório para arquivos de log da sessão de jogo.

Tipo: `Aws::GameLift::Server::LogParameters`

Obrigatório: não

onStartGameSession

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para ativar uma nova sessão de jogo. O Amazon GameLift chama essa função em resposta à solicitação do cliente [CreateGameSession](#). A função de retorno de chamada passa um objeto [GameSession](#) (definido na Referência de API do serviço do Amazon GameLift).

Digite: `const std::function<void(Aws::GameLift::Model::GameSession)>
onStartGameSession`

Obrigatório: Sim

onProcessTerminate

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para forçar o desligamento do processo de servidor. Depois de chamar essa função, o Amazon GameLift aguardará cinco minutos para o processo de servidor encerrar e responder com uma chamada [ProcessEnding\(\)](#). Se nenhuma resposta for recebida, ele desativará o processo do servidor.

Digite: `std::function<void()> onProcessTerminate`

Obrigatório: não

onHealthCheck

Nome da função de retorno que o serviço do Amazon GameLift invoca para solicitar um relatório de status de integridade do processo de servidor. O Amazon GameLift chama essa função a cada 60 segundos. Depois de chamar essa função, o Amazon GameLift aguardará 60 segundos por uma resposta e, se nenhuma for recebida, registra o processo do servidor como não íntegro.

Digite: `std::function<bool()> onHealthCheck`

Obrigatório: não

`onUpdateGameSession`

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. O Amazon GameLift chama essa função quando uma solicitação de [alocação de correspondência](#) é processada para fornecer dados atualizados do marcador de jogos. Ele passa um objeto [GameSession](#), uma atualização de status (`updateReason`) e o ID do tíquete de alocação de correspondência.

Digite:

```
std::function<void(Aws::GameLift::Server::Model::UpdateGameSession)>  
onUpdateGameSession
```

Obrigatório: não

`StartMatchBackfillRequest`

Esse tipo de dados é usado para enviar uma solicitação de alocação de marcação de jogos. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StartMatchBackfill\(\)](#).

Índice

`GameSessionArn`

Identificador da sessão de jogo exclusivo. A ação de API [GetGameSessionId\(\)](#) retorna o identificador no formato de ARN.

Tipo: sequência

Obrigatório: Sim

`MatchmakingConfigurationArn`

O identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. Para encontrar o marcador de jogos usado para criar a sessão de jogo original, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Saiba mais sobre os dados do marcador de jogos no [Trabalhar com dados do marcador de jogos](#).

Tipo: sequência

Obrigatório: Sim

Jogadores

Um conjunto de dados que representa todos os jogadores que estão na sessão do jogo no momento. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais. Consulte o Guia de Referência de API do Amazon GameLift para obter uma descrição do formato do objeto Player. Para encontrar atributos, IDs e atribuições de equipe dos jogadores, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Se o marcador de jogos usa latência, obtenha a latência atualizada para a região atual e a inclua nos dados de cada jogador.

Tipo: `std::vector<player>`

Obrigatório: Sim

TicketId

O identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se nenhum valor for fornecido aqui, o Amazon GameLift gerará um no formato de UUID. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.

Tipo: sequência

Obrigatório: não

StopMatchBackfillRequest

Esse tipo de dados é usado para cancelar uma solicitação de alocação de marcação de jogo. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StopMatchBackfill\(\)](#).

Índice

GameSessionArn

O identificador de sessão de jogo exclusivo associado à solicitação que está sendo cancelada.

Tipo: sequência

Obrigatório: Sim

MatchmakingConfigurationArn

O identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada.

Tipo: sequência

Obrigatório: Sim

TicketId

O identificador exclusivo do tíquete de solicitação de alocação a ser cancelado.

Tipo: sequência

Obrigatório: Sim

Referência do SDK do servidor do Amazon GameLift para C#

Você pode usar essa referência do SDK do servidor Amazon GameLift C# para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)
- [Referência 4.x do SDK do servidor Amazon GameLift para C#](#)

Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity

Você poderá usar essa referência do SDK 5.x do servidor C# do Amazon GameLift para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in do SDK do servidor C# para Unity, consulte [Integre o Amazon GameLift em um projeto do Unity](#). O SDK 5.x do servidor do Amazon GameLift para C# oferece suporte ao .NET 4.6 e .NET 6.

Tópicos

- [Referência GameLift do SDK do servidor Amazon para C# e Unity: ações](#)
- [Referência do SDK do servidor Amazon GameLift para C# e Unity: tipos de dados](#)

Referência GameLift do SDK do servidor Amazon para C# e Unity: ações

Essa referência do SDK do servidor Amazon GameLift C# ajuda você a preparar seu jogo multijogador para uso com a Amazon. GameLift Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in do SDK do servidor C# para Unity, consulte [Integre o Amazon GameLift em um projeto do Unity](#).

Ações

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetTerminationTime\(\)](#)
- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)
- [Destroy\(\)](#)

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
AwsStringOutcome GetSdkVersion();
```

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto [the section called “AwsStringOutcome”](#). A string retornada inclui o número da versão (exemplo, 5.0.0). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
var getSdkVersionOutcome = GameLiftServerAPI.GetSdkVersion();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma frota EC2 gerenciada. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método lê os parâmetros do servidor do ambiente host para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
GenericOutcome InitSDK();
```

Valor de retorno

Se for bem-sucedido, retornará um `InitSdkOutcome` objeto para indicar que o processo do servidor está pronto para ser chamado [ProcessReady\(\)](#).

Exemplo

```
//Call InitSDK to establish a local connection with the GameLift agent to enable  
further communication.  
GenericOutcome initSDKOutcome = GameLiftServerAPI.InitSDK();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma Anywhere frota. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método requer parâmetros explícitos do servidor para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
GenericOutcome InitSDK(ServerParameters serverParameters);
```

Parâmetros

[Parâmetros do servidor](#)

Para inicializar um servidor de jogo em uma GameLift Anywhere frota da Amazon, construa um `ServerParameters` objeto com as seguintes informações:

- O URL do WebSocket usado para se conectar ao seu servidor de jogo.
- O ID do processo usado para hospedar o servidor de jogos.
- O ID do computador que hospeda os processos do seu servidor de jogos.
- O ID da GameLift frota da Amazon que contém sua GameLift Anywhere computação da Amazon.
- O token de autorização gerado pela GameLift operação da Amazon.

Valor de retorno

Se for bem-sucedido, retornará um `InitSdkOutcome` objeto para indicar que o processo do servidor está pronto para ser chamado [ProcessReady\(\)](#).

Note

Se as chamadas para `InitSDK()` estiverem falhando para compilações de jogos implantadas em frotas do Anywhere, verifique o parâmetro `ServerSdkVersion` usado ao criar o recurso de compilação. Você deve definir explicitamente esse valor para a versão do SDK do servidor em uso. O valor padrão desse parâmetro é 4.x, o que não é compatível. Para resolver esse problema, crie uma nova versão e implante-a em uma nova frota.

Exemplo

```
//Define the server parameters
string websocketUrl = "wss://us-west-1.api.amazongamelift.com";
string processId = "PID1234";
string fleetId = "aarn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
```

```
string hostId = "HardwareAnywhere";
string authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
ServerParameters serverParameters =
    new ServerParameters(webSocketUrl, processId, hostId, fleetId, authToken);

//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
GenericOutcome initSDKOutcome = GameLiftServerAPI.InitSDK(serverParameters);
```

ProcessReady()

Notifica a Amazon de GameLift que o processo do servidor está pronto para hospedar sessões de jogo. Chame esse método após invocar [InitSDK\(\)](#). Esse método deve ser chamado somente uma vez por processo.

Sintaxe

```
GenericOutcome ProcessReady(ProcessParameters processParameters)
```

Parâmetros

[ProcessParameters](#)

Um objeto `ProcessParameters` contém informações sobre o processo do servidor.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra as implementações do método e da função delegada.

```
// Set parameters and call ProcessReady
ProcessParameters processParams = new ProcessParameters(
    this.OnStartGameSession,
    this.OnProcessTerminate,
    this.OnHealthCheck,
    this.OnUpdateGameSession,
    port,
    new LogParameters(new List<string>())
    // Examples of log and error files written by the game server
```

```
{
    "C:\\game\\logs",
    "C:\\game\\error"
})
);
GenericOutcome processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);
```

ProcessEnding()

Notifica a Amazon de GameLift que o processo do servidor está sendo encerrado. Chame esse método depois de todas as outras tarefas de limpeza (incluindo o encerramento da sessão ativa do jogo) e antes de encerrar o processo. Dependendo do resultado `ProcessEnding()`, o processo sai com êxito (0) ou erro (-1) e gera um evento de frota. Se o processo for encerrado com um erro, o evento de frota gerado será `SERVER_PROCESS_TERMINATED_UNHEALTHY`.

Sintaxe

```
GenericOutcome ProcessEnding()
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo chama `ProcessEnding()` e `Destroy()` antes de encerrar o processo do servidor com um código de saída bem-sucedido ou de erro.

```
GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();
GameLiftServerAPI.Destroy();

if (processEndingOutcome.Success)
{
    Environment.Exit(0);
}
else
{
    Console.WriteLine("ProcessEnding() failed. Error: " +
        processEndingOutcome.Error.ToString());
    Environment.Exit(-1);
}
```

ActivateGameSession()

Notifica a Amazon de GameLift que o processo do servidor ativou uma sessão de jogo e agora está pronto para receber conexões de jogadores. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, após toda inicialização da sessão do jogo.

Sintaxe

```
GenericOutcome ActivateGameSession()
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo mostra `ActivateGameSession()` sendo chamado como parte da função de delegação `onStartGameSession()`.

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

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador.

Sintaxe

```
GenericOutcome UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy
playerSessionPolicy)
```

Parâmetros

playerSessionPolicy

Valor de string que indica se a sessão do jogo aceita novos jogadores.

Os valores válidos são:

- ACCEPT_ALL – Aceite todas as novas sessões de jogador.
- DENY_ALL – Recuse todas as novas sessões de jogador.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
GenericOutcome updatePlayerSessionPolicyOutcome =  
    GameLiftServerAPI.UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy.ACCEPT_ALL);
```

GetGameSessionId()

Recupera o ID da sessão de jogo hospedada pelo processo do servidor ativo.

Para processos inativos que não são ativados com uma sessão de jogo, a chamada retorna um [the section called "GameLiftError"](#).

Sintaxe

```
AwsStringOutcome GetGameSessionId()
```

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto [the section called "AwsStringOutcome"](#). Se não for bem-sucedido, retornará uma mensagem de erro."

Exemplo

```
AwsStringOutcome getGameSessionIdOutcome = GameLiftServerAPI.GetGameSessionId();
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado, se essa informação estiver disponível. Um processo de servidor executa essa ação após receber um `onProcessTerminate()` retorno de chamada da Amazon GameLift. A Amazon GameLift liga `onProcessTerminate()` pelos seguintes motivos:

- Quando o processo do servidor relatou problemas de saúde ou não respondeu à Amazon GameLift.
- Ao encerrar a instância durante um evento de redução.
- Quando uma instância é encerrada devido a uma [interrupção na instância spot](#).

Sintaxe

```
AwsDateTimeOutcome GetTerminationTime()
```

Valor de retorno

Se for bem-sucedido, retornará o horário de término como um objeto [the section called "AwsDateTimeOutcome"](#). O valor é o tempo de rescisão, expresso em tiques decorridos desde então 0001 00:00:00. Por exemplo, o valor da data e hora 2020-09-13 12:26:40 -000Z é igual aos tiques 637355968000000000. Se nenhum horário de rescisão estiver disponível, o retornará uma mensagem de erro.

Exemplo

```
AwsDateTimeOutcome getTerminationTimeOutcome = GameLiftServerAPI.GetTerminationTime();
```

AcceptPlayerSession()

Notifica a Amazon de GameLift que um jogador com o ID de sessão de jogador especificado se conectou ao processo do servidor e precisa de validação. A Amazon GameLift verifica se o ID da sessão do jogador é válido. Depois que a sessão do jogador é validada, a Amazon GameLift altera o status do slot do jogador de RESERVADO para ATIVO.

Sintaxe

```
GenericOutcome AcceptPlayerSession(String playerSessionId)
```

Parâmetros

playerSessionId

ID exclusivo emitido GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra uma função para processar uma solicitação de conexão, inclusive validando e rejeitando IDs de sessão do jogador inválidos.

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerSessionId)
{
    GenericOutcome acceptPlayerSessionOutcome =
    GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
    if(acceptPlayerSessionOutcome.Success)
    {
        connectionToSessionMap.emplace(connection, playerSessionId);
        connection.Accept();
    }
    else
    {
        connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);
    }
}
```

RemovePlayerSession()

Notifica a Amazon de GameLift que um jogador se desconectou do processo do servidor. Em resposta, a Amazon GameLift altera o slot do jogador para disponível.

Sintaxe

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
GenericOutcome removePlayerSessionOutcome =  
    GameLiftServerAPI.RemovePlayerSession(playerSessionId);
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, que incluem configurações, metadados da sessão e dados do jogador. Use essa ação para obter informações de uma única sessão de jogador, para todas as sessões de jogador em uma sessão de jogo, ou para todas as sessões de jogador associadas a um único ID de jogador.

Sintaxe

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest  
    describePlayerSessionsRequest)
```

Parâmetros

[DescribePlayerSessionsRequest](#)

Um objeto [the section called "DescribePlayerSessionsRequest"](#) que descreve quais sessões de jogador recuperar.

Valor de retorno

Se bem-sucedido, retorna um objeto [the section called "Descreva o resultado das sessões de jogadores"](#) que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação.

Exemplo

Este exemplo ilustra uma solicitação para todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Ao omitir NextToken e definir o valor limite como 10, a Amazon GameLift retornará os primeiros registros de sessão de 10 jogadores que correspondam à solicitação.

```
// Set request parameters  
DescribePlayerSessionsRequest describePlayerSessionsRequest = new  
    DescribePlayerSessionsRequest()
```

```
{
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    //gets the ID for the
current game session
    Limit = 10,
    PlayerSessionStatusFilter =
        PlayerSessionStatusMapper.GetNameForPlayerSessionStatus(PlayerSessionStatus.ACTIVE)
};
// Call DescribePlayerSessions
DescribePlayerSessionsOutcome describePlayerSessionsOutcome =
    GameLiftServerAPI.DescribePlayerSessions(describePlayerSessionsRequest);
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Esta ação é assíncrona. Se novos jogadores forem combinados, a Amazon GameLift fornece dados atualizados do matchmaker usando a função de retorno de chamada. `OnUpdateGameSession()`

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest
startBackfillRequest);
```

Parâmetros

[StartMatchBackfillRequest](#)

Um objeto `StartMatchBackfillRequest` contém informações sobre a solicitação de alocação.

Valor de retorno

Retorna um objeto [the section called “StartMatchBackfillOutcome”](#) com o ID do tíquete de alocação de correspondência ou um falha com uma mensagem de erro.

Exemplo

```
// Build a backfill request
StartMatchBackfillRequest startBackfillRequest = new StartMatchBackfillRequest()
{
    TicketId = "1111aaaa-22bb-33cc-44dd-5555eeee66ff", //optional
    MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig",
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result, // gets ID for
current game session
    MatchmakerData matchmakerData =
    MatchmakerData.FromJson(gameSession.MatchmakerData), // gets matchmaker data for
current players
    // get matchmakerData.Players
    // remove data for players who are no longer connected
    Players = ListOfPlayersRemainingInTheGame
};

// Send backfill request
StartMatchBackfillOutcome startBackfillOutcome =
    GameLiftServerAPI.StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void OnUpdateGameSession(GameSession myGameSession)
{
    // game-specific tasks to prepare for the newly matched players and update matchmaker
data as needed
}
```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Sintaxe

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parâmetros

[StopMatchBackfillRequest](#)

Um objeto `StopMatchBackfillRequest` que fornece detalhes sobre o tíquete de marcação de jogos que você está interrompendo.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set backfill stop request parameters
StopMatchBackfillRequest stopBackfillRequest = new StopMatchBackfillRequest(){
    TicketId = "1111aaaa-22bb-33cc-44dd-5555eeee66ff", //optional, if not provided one is
    autogenerated
    MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig",
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result //gets the ID for the
    current game session
};
GenericOutcome stopBackfillOutcome =
    GameLiftServerAPI.StopMatchBackfillRequest(stopBackfillRequest);
```

`GetComputeCertificate()`

Recupera o caminho para o certificado TLS usado para criptografar a conexão de rede entre o servidor do jogo e o cliente do jogo. Você pode usar o caminho do certificado ao registrar seu dispositivo computacional em uma GameLift Anywhere frota da Amazon. Para obter mais informações, consulte, [RegisterCompute](#).

Sintaxe

```
GetComputeCertificateOutcome GetComputeCertificate();
```

Valor de retorno

Retorna um `GetComputeCertificateResponse` objeto que contém o seguinte:

- `CertificatePath`: o caminho para o certificado TLS em seu recurso computacional. Ao usar uma frota GameLift gerenciada pela Amazon, esse caminho contém:

- `certificate.pem`: o certificado do usuário final. A cadeia completa de certificados é a combinação de `certificateChain.pem` anexados a esse certificado.
- `certificateChain.pem`: a cadeia de certificados que contém o certificado raiz e os certificados intermediários.
- `rootCertificate.pem`: Certificado raiz.
- `privateKey.pem`: a chave privada para o certificado do usuário final.
- `ComputeName`: o nome do seu recurso computacional.

Exemplo

```
GetComputeCertificateOutcome getComputeCertificateOutcome =  
    GameLiftServerAPI.GetComputeCertificate();
```

GetFleetRoleCredentials()

Recupera as credenciais da função do IAM que autorizam a GameLift Amazon a interagir com outras. Serviços da AWS Para ter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Sintaxe

```
GetFleetRoleCredentialsOutcome GetFleetRoleCredentials(GetFleetRoleCredentialsRequest  
    request);
```

Parâmetros

[Obtenha a solicitação de credenciais do FleetRole](#)

Credenciais de função que ampliam o acesso limitado aos seus recursos AWS no servidor do jogo.

Valor de retorno

Informa um objeto [the section called "GetFleetRoleCredentialsOutcome"](#).

Exemplo

```
// form the fleet credentials request
```

```
GetFleetRoleCredentialsRequest getFleetRoleCredentialsRequest = new
    GetFleetRoleCredentialsRequest(){
        RoleArn = "arn:aws:iam::123456789012:role/service-role/exampleGameLiftAction"
    };
GetFleetRoleCredentialsOutcome GetFleetRoleCredentialsOutcome credentials =
    GetFleetRoleCredentials(getFleetRoleCredentialsRequest);
```

Destroy()

Libera o SDK do servidor de GameLift jogos da Amazon da memória. Como prática recomendada, chame esse método antes `ProcessEnding()` e depois de encerrar o processo. Se você estiver usando uma frota Anywhere e não estiver encerrando os processos do servidor após cada sessão de jogo, ligue `Destroy()` e reinicialize antes de notificar `InitSDK()` a Amazon de GameLift que o processo está pronto para hospedar uma sessão de jogo. `ProcessReady()`

Sintaxe

```
GenericOutcome Destroy()
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Operations to end game sessions and the server process
GenericOutcome processEndingOutcome = GameLiftServerAPI.ProcessEnding();

// Shut down and destroy the instance of the GameLift Game Server SDK
GenericOutcome destroyOutcome = GameLiftServerAPI.Destroy();

// Exit the process with success or failure
if (processEndingOutcome.Success)
{
    Environment.Exit(0);
}
else
{
    Console.WriteLine("ProcessEnding() failed. Error: " +
        processEndingOutcome.Error.ToString());
    Environment.Exit(-1);
}
```



```
}
```

Referência do SDK do servidor Amazon GameLift para C# e Unity: tipos de dados

Esta referência do Server SDK C# do Amazon GameLift pode ajudá-lo a preparar o jogo multijogador a ser usado com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in do SDK do servidor C# para Unity, consulte [Integre o Amazon GameLift em um projeto do Unity](#).

Tipos de dados

- [LogParameters](#)
- [ProcessParameters](#)
- [UpdateGameSession](#)
- [GameSession](#)
- [Parâmetros do servidor](#)
- [StartMatchBackfillRequest](#)
- [Jogador](#)
- [DescribePlayerSessionsRequest](#)
- [StopMatchBackfillRequest](#)
- [Obtenha a solicitação de credenciais do FleetRole](#)
- [AttributeValue](#)
- [AwsStringOutcome](#)
- [Resultado genérico](#)
- [Descreva o resultado das sessões de jogadores](#)
- [Descreva o resultado das sessões de jogadores](#)
- [PlayerSession](#)
- [StartMatchBackfillOutcome](#)
- [StartMatchBackfillResult](#)
- [GetComputeCertificateOutcome](#)
- [Obter resultado do certificado de computação](#)
- [GetFleetRoleCredentialsOutcome](#)
- [GetFleetRoleCredentialsResult](#)

- [AwsDateTimeOutcome](#)
- [GameLiftError](#)
- [Enumerações](#)

LogParameters

Use esse tipo de dados para identificar quais arquivos gerados durante uma sessão de jogo você deseja que o servidor do jogo o faça upload no Amazon GameLift após o término da sessão do jogo. O servidor do jogo comunica `LogParameters` to o Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Properties	Descrição
LogPaths	<p>A lista de caminhos de diretório para arquivos de log do servidor de jogo que você deseja que o Amazon GameLift armazene para acesso futuro. O processo do servidor gera esses arquivos durante cada sessão de jogo. Você define caminhos e nomes de arquivos em seu servidor de jogo e os armazena no diretório raiz de construção do jogo.</p> <p>Os caminhos de log devem ser absolutos. Por exemplo, se a compilação do jogo armazenasse logs de sessão de jogo em um caminho como <code>MyGame\sessionLogs\</code>, o caminho estará <code>c:\game\MyGame\sessionLogs</code> em uma instância do Windows.</p> <p>Digite: <code>List<String></code></p> <p>Obrigatório: não</p>

ProcessParameters

Este tipo de dados contém o conjunto de parâmetros enviado para o serviço da Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Properties	Descrição
LogParameters	<p>O objeto com uma lista de caminhos de diretório para arquivos de log da sessão de jogo.</p> <p>Digite: <code>Aws::GameLift::Server:: LogParameters</code></p> <p>Obrigatório: Sim</p>
OnHealthCheck	<p>O nome da função de retorno de chamada que o Amazon GameLift invoca para solicitar um relatório de status de integridade do processo do servidor. O Amazon GameLift chama essa função a cada 60 segundos. Depois de chamar essa função, o Amazon GameLift aguardará 60 segundos por uma resposta, se nenhuma for recebida, o Amazon GameLift registra o processo do servidor como não íntegro.</p> <p>Digite: <code>void OnHealthCheckDelegate()</code></p> <p>Obrigatório: Sim</p>
OnProcessTerminate	<p>O nome da função de retorno de chamada que o Amazon GameLift invoca para forçar o encerramento do processo do servidor. Depois de chamar essa função, o Amazon GameLift aguardará cinco minutos para o processo de servidor desligar e responder com uma chamada ProcessEnding() antes do desligamento do processo de servidor.</p> <p>Digite: <code>void OnProcessTerminateDelegate()</code></p> <p>Obrigatório: Sim</p>

OnStartGameSession

O nome da função de retorno de chamada que o Amazon GameLift invoca para ativar uma nova sessão de jogo. O Amazon GameLift chama essa função em resposta à solicitação do cliente [CreateGameSession](#). A função de retorno de chamada usa um objeto [GameSession](#) definido na referência da API do Amazon GameLift.

Digite: `void OnStartGameSession
Delegate(GameSession)`

Obrigatório: Sim

OnUpdateGameSession

O nome da função de retorno de chamada que o Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. O Amazon GameLift chama essa função quando uma solicitação de preenchimento de correspondência é processada para fornecer dados atualizados do marcador de jogos. Ele passa um objeto [GameSession](#), uma atualização de status (`updateReason`) e o ID do tíquete de alocação de correspondência.

Tipo: anular `OnUpdateGameSessionDelegate
(UpdateGameSession)`

Obrigatório: não

<p>Porta</p>	<p>O número da porta em que o processo de servidor escuta para novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.</p> <p>Digite: Integer</p> <p>Obrigatório: Sim</p>
---------------------	---

UpdateGameSession

As informações atualizadas de um objeto de sessão de jogo incluem o motivo pelo qual a sessão de jogo foi atualizada. Se a atualização estiver relacionada a uma ação de alocação de correspondência, esse tipo de dados inclui o ID do tíquete de alocação.

Propriedades	Descrição
<p>GameSession</p>	<p>Um objeto GameSession definido pela API Amazon GameLift. O objeto GameSession contém propriedades que descrevem uma sessão de jogo.</p> <p>Digite: GameSession GameSession()</p> <p>Obrigatório: Sim</p>
<p>UpdateReason</p>	<p>O motivo pelo qual a sessão do jogo está sendo atualizada.</p> <p>Digite: UpdateReason UpdateReason()</p> <p>Obrigatório: Sim</p>

Propriedades	Descrição
BackfillTicketId	<p>O ID do tíquete de alocação que está tentando atualizar a sessão do jogo.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>

GameSession

Detalhes de uma sessão de jogo.

Propriedades	Descrição
GameSessionId	<p>Um identificador exclusivo da sessão de jogo. Um ARN de sessão de jogo tem o seguinte formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> .</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
Nome	<p>Um rótulo descritivo da sessão do jogo.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
FleetId	<p>Um identificador exclusivo da frota em que a sessão de jogo está sendo executada.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
MaximumPlayerSessionCount	<p>O número máximo de conexões do jogador com a sessão do jogo.</p> <p>Digite: Integer</p> <p>Obrigatório: não</p>
Porta	<p>O número da porta da sessão de jogo. Para se conectar a um servidor de jogos Amazon GameLift, um aplicativo precisa do endereço IP e do número da porta.</p> <p>Digite: Integer</p> <p>Obrigatório: não</p>
IpAddress	<p>O endereço IP da sessão de jogo. Para se conectar a um servidor de jogos Amazon GameLift, um aplicativo precisa do endereço IP e do número da porta.</p> <p>Digite: String</p> <p>Obrigatório: não</p>
GameSessionData	<p>Um conjunto de propriedades de sessão de jogo personalizadas, formatadas como um único valor de string.</p> <p>Digite: String</p> <p>Obrigatório: não</p>

Propriedades	Descrição
MatchmakerData	<p>As informações sobre o processo de marcação de jogos usado para criar a sessão do jogo, na sintaxe JSON, formatado como uma string. Além da configuração de marcação de jogos usada, ela contém dados sobre todos os jogadores designados para a partida, incluindo atributos do jogador e atribuições da equipe.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
GameProperties	<p>Um conjunto de propriedades personalizadas para uma sessão de jogo, formatado como pares de chave:valor. Essas propriedades são passadas com uma solicitação de início de uma nova sessão de jogo.</p> <p>Digite: <code>Dictionary<string, string></code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
DnsName	<p>O identificador DNS atribuído à instância que está executando a sessão do jogo. Os valores têm o formato a seguir:</p> <ul style="list-style-type: none"> Frotas habilitadas para TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> Frotas não habilitadas para TLS: <code>ec2-unique identifier>.compute.amazonaws.com</code> <p>Ao se conectar a uma sessão de jogo que está sendo executada em uma frota compatível com TLS, você deve usar o nome DNS, não o endereço IP.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

Parâmetros do servidor

Informações usadas para manter a conexão entre um servidor Anywhere do Amazon GameLift e o serviço do Amazon GameLift. Essas informações são usadas ao iniciar novos processos de servidor com [InitSDK\(\)](#). Para servidores hospedados em instâncias EC2 gerenciadas pelo Amazon GameLift, use um objeto vazio.

Propriedades	Descrição
WebSocketUrl	<p>O <code>GameLiftServerSdkEndpoint</code> retornado quando você usa o <code>RegisterCompute</code> como parte do Anywhere do Amazon GameLift.</p> <p>Digite: <code>String</code></p>

Propriedades	Descrição
	Obrigatório: Sim
ProcessId	<p>Um identificador exclusivo registrado no processo do servidor que hospeda seu jogo.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>
HostId	<p>Um identificador exclusivo para o host com os processos do servidor que hospedam seu jogo. O HOSTID é o computeName usado quando você registrou seu computador. Para obter mais informações, consulte RegisterCompute</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>
FleetId	<p>O ID da frota na qual o computador está registrado. Para obter mais informações, consulte RegisterCompute.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>
AuthToken	<p>O token de autenticação gerado pelo Amazon GameLift que autentica seu servidor no Amazon GameLift. Para obter mais informações, consulte GetComputeAuthToken.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>

StartMatchBackfillRequest

Informações usadas para criar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações ao Amazon GameLift em uma chamada [StartMatchBackfill\(\)](#).

Propriedades	Descrição
GameSessionArn	<p>O identificador da sessão de jogo exclusivo . A operação de API retorna GetGameSessionId o identificador no formato de ARN.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>
MatchmakingConfigurationArn	<p>O identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. O marcador de jogos ARN da sessão de jogo original está no objeto de sessão de jogo na propriedade de dados do marcador de jogos. Saiba mais sobre os dados do marcador de jogos em Trabalhar com dados do marcador de jogos.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: Sim</p>
Jogadores	<p>Um conjunto de dados que representa todos os jogadores que estão na sessão do jogo no momento. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais.</p> <p>Digite: <code>List<Player></code></p> <p>Obrigatório: Sim</p>

Propriedades	Descrição
TicketId	<p>O identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se você não fornecer um valor, o Amazon GameLift o gerará um. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

Jogador

Representa um jogador em marcação de jogos. Quando uma solicitação de marcação de jogos começa, um jogador tem um ID de jogador, atributos, e possivelmente dados de latência. O Amazon GameLift adiciona informações da equipe após a realização de uma partida.

Propriedades	Descrição
Latência em MS	<p>Um conjunto de valores expressos em milissegundos, que indicam a quantidade de latência que um jogador experimenta quando conectado a um local.</p> <p>Se essa propriedade for usada, o player só será correspondido aos locais listados. Se um marcador de jogos tiver uma regra que avalie a latência do jogador, os jogadores deverão reportar a latência a ser correspondida.</p> <p>Digite: <code>Dictionary<string, int></code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
Atributos do jogador	<p>Uma coleção de pares de chave:valor que contêm informações de jogadores para uso em marcação de jogos. As chaves de atributos do jogador devem corresponder aos <code>PlayerAttributes</code> usados em um conjunto de regras de marcação de jogos.</p> <p>Para obter mais informações sobre os atributos de jogador, consulte AttributeValue.</p> <p>Digite: <code>Dictionary<string, AttributeValue></code></p> <p>Obrigatório: não</p>
PlayerId	<p>Um identificador exclusivo para um jogador.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
Team	<p>O nome da equipe à qual o jogador está designado em uma partida. Você define o nome da equipe no conjunto de regras de marcação de jogos.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

DescribePlayerSessionsRequest

Esse tipo de dados é usado para especificar quais sessões de jogador recuperar. Ele pode ser usado de várias maneiras: (1) fornecer um `PlayerSessionId` para solicitar uma sessão de jogador específica; (2) fornecer um `GameSessionId` para solicitar todas as sessões de jogador na sessão de jogo especificada ou (3) fornecer um `PlayerId` para solicitar todas as sessões de jogador para

o jogador. Para coleções maiores de sessões de jogador, use os parâmetros de paginação para recuperar resultados como páginas sequenciais.

Propriedades	Descrição
GameSessionId	<p>O identificador da sessão de jogo exclusivo . Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada. O formato do ID da sessão de jogo é o seguinte: <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string></code> . O valor de <code><ID string></code> é uma string de ID personalizada (se uma tiver sido especificada quando a sessão do jogo foi criada) ou uma string gerada.</p> <p>Digite: String</p> <p>Obrigatório: não</p>
PlayerSessionId	<p>O Identificador exclusivo para uma sessão de jogador.</p> <p>Digite: String</p> <p>Obrigatório: não</p>
PlayerId	<p>O identificador exclusivo para um jogador. Consulte Gerar IDs de jogador.</p> <p>Digite: String</p> <p>Obrigatório: não</p>
PlayerSessionStatusFilter	<p>O status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis estão os seguintes:</p> <ul style="list-style-type: none"> RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda

Propriedades	Descrição
	<p>não se conectou ao processo de servidor e/ou foi validado.</p> <ul style="list-style-type: none">• ACTIVE – O jogador foi validado pelo processo de servidor e está conectado no momento.• COMPLETED – A conexão do jogador foi interrompida.• TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou e/ou não foi validado dentro do tempo limite (60 segundos). <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
NextToken	<p>O token indicando o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
Limite	<p>O número máximo de resultados a serem retornados. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>

StopMatchBackfillRequest

Informações usadas para cancelar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações ao serviço Amazon GameLift em uma chamada [StopMatchBackfill\(\)](#).

Propriedades	Descrição
GameSessionArn	O identificador de sessão de jogo exclusivo da solicitação que está sendo cancelada. Digite: <code>string</code> Obrigatório: Sim
MatchmakingConfigurationArn	O identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada. Digite: <code>string</code> Obrigatório: Sim
TicketId	O identificador exclusivo do tíquete de solicitação de alocação a ser cancelado. Digite: <code>string</code> Obrigatório: Sim

Obtenha a solicitação de credenciais do FleetRole

Esse tipo de dados dá ao servidor do jogo acesso limitado aos seus outros recursos AWS. Para obter mais informações, consulte, [Configurar uma função de serviço do IAM para a Amazon GameLift](#).

Propriedades	Descrição
RoleArn	O nome do recurso da Amazon (ARN) do perfil de serviço que estende o acesso limitado aos seus recursos AWS.

Propriedades	Descrição
	Digite: <code>string</code> Obrigatório: Sim
RoleSessionName	O nome da sessão que descreve o uso das credenciais da função. Digite: <code>string</code> Obrigatório: não

AttributeValue

Use esses valores em pares de chave-valor de atributo [Jogador](#). Esse objeto permite especificar um valor de atributo usando qualquer um dos tipos de dados válidos: `string`, número, matriz de `string` ou mapa de dados. Cada objeto `AttributeValue` pode usar somente uma das propriedades disponíveis.

Propriedades	Descrição
attrType	Especifica o tipo de valor do atributo. Tipo: Um <code>AttrType</code> valor enum .. Obrigatório: não
S	Representa um valor de atributo de <code>string</code> . Digite: <code>string</code> Obrigatório: Sim
N	Representa um valor de atributo numérico. Digite: <code>double</code> Obrigatório: Sim

Propriedades	Descrição
SL	<p>Representa uma matriz de valores de atributos de string.</p> <p>Digite: <code>string[]</code></p> <p>Obrigatório: Sim</p>
SDM	<p>Representa um dicionário de chaves de string e valores duplos.</p> <p>Digite: <code>Dictionary<string, double></code></p> <p>Obrigatório: Sim</p>

AwsStringOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Result	<p>O resultado da ação.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called "GameLiftError"</p> <p>Obrigatório: não</p>

Resultado genérico

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called "GameLiftError"</p> <p>Obrigatório: não</p>

Descreva o resultado das sessões de jogadores

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	<p>O resultado da ação.</p> <p>Digite: the section called "Descreva o resultado das sessões de jogadores"</p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p>

Propriedades	Descrição
	<p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

Descreva o resultado das sessões de jogadores

Propriedades	Descrição
NextToken	<p>O token indicando o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: Sim</p>
Sessões de jogadores	<p>Uma coleção de objetos contendo propriedades para cada sessão de jogador que corresponde à solicitação.</p> <p>Digite: <code>IList<the section called “PlayerSession”></code></p> <p>Obrigatório:</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p>

Propriedades	Descrição
	Obrigatório: não

PlayerSession

Propriedades	Descrição
CreationTime	Digite: long Obrigatório: Sim
FleetId	Digite: string Obrigatório: Sim
GameSessionId	Digite: string Obrigatório: Sim
IpAddress	Digite: string Obrigatório: Sim
Dados do jogador	Digite: string Obrigatório: Sim
PlayerId	Digite: string Obrigatório: Sim
PlayerSessionId	Digite: string Obrigatório: Sim
Porta	Digite: int Obrigatório: Sim
Status	Tipo: Um PlayerSessionStatus enum..

Propriedades	Descrição
	Obrigatório: Sim
Hora de rescisão	Digite: long Obrigatório: Sim
DnsName	Digite: string Obrigatório: Sim

StartMatchBackfillOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “StartMatchBackfillResult” Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “GameLiftError” Obrigatório: não

StartMatchBackfillResult

Propriedades	Descrição
TicketId	Digite: <code>string</code> Obrigatório: Sim

GetComputeCertificateOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “Obter resultado do certificado de computação” Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “GameLiftError” Obrigatório: não

Obter resultado do certificado de computação

O caminho para o certificado TLS em seu computador e o nome do host do computador.

Propriedades	Descrição
CertificatePath	Digite: <code>string</code> Obrigatório: Sim
ComputeName	Digite: <code>string</code> Obrigatório: Sim

GetFleetRoleCredentialsOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “GetFleetRoleCredentialsResult” Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “GameLiftError” Obrigatório: não

GetFleetRoleCredentialsResult

Propriedades	Descrição
AccessKeyId	<p>O ID da chave de acesso para autenticar e fornecer acesso aos seus recursos AWS.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
AssumedRoleId	<p>O ID do usuário ao qual o perfil de serviço pertence.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
AssumedRoleUserArn	<p>O nome do recurso da Amazon (ARN) do usuário ao qual pertence o perfil de serviço.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
Expiração	<p>A quantidade de tempo até que suas credenciais de sessão expirem.</p> <p>Digite: <code>DateTime</code></p> <p>Obrigatório: não</p>
SecretAccessKey	<p>O ID da chave de acesso secreta para autenticação.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
SessionToken	<p>Um token para identificar a sessão ativa atual interagindo com seus recursos AWS.</p>

Propriedades	Descrição
	Digite: <code>string</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError" Obrigatório: não

AwsDateTimeOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Result	O resultado da ação. Digite: <code>DateTime</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code> Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called "GameLiftError"

Propriedades	Descrição
	Obrigatório: não

GameLiftError

Propriedades	Descrição
ErrorType	O tipo de erro. Tipo: Um GameLiftErrorType enum . Obrigatório: não
Nome do erro	O nome do erro. Digite: <code>string</code> Obrigatório: não
ErrorMessage	A mensagem de erro. Digite: <code>string</code> Obrigatório: não

Enumerações

As enumerações definidas para o SDK do servidor (C#) do Amazon GameLift são definidos da seguinte forma:

AttrType

- NONE
- STRING
- DOUBLE
- STRING_LIST
- STRING_DOUBLE_MAP

Tipo de erro do GameLift

Valor da string indicando o tipo de erro. Os valores válidos são:

- `SERVICE_CALL_FAILED` – Uma chamada AWS para um serviço falhou.
- `LOCAL_CONNECTION_FAILED` – A conexão local com o Amazon GameLift falhou.
- `NETWORK_NOT_INITIALIZED` – A rede não foi inicializada.
- `GAMESESSION_ID_NOT_SET` – O ID da sessão do jogo não foi definido.
- `BAD_REQUEST_EXCEPTION`
- `INTERNAL_SERVICE_EXCEPTION`
- `ALREADY_INITIALIZED` – servidor ou cliente Amazon GameLift já foi inicializado com `Initialize()`.
- `FLEET_MISMATCH` – A frota de destino não corresponde à frota de uma `GameSession` ou `PlayerSession`.
- `GAMELIFT_CLIENT_NOT_INITIALIZED` – O cliente Amazon GameLift não foi inicializado.
- `GAMELIFT_SERVER_NOT_INITIALIZED` – O servidor Amazon GameLift não foi inicializado.
- `GAME_SESSION_ENDED_FAILED` — O Amazon GameLift Server SDK não conseguiu entrar em contato com o serviço para informar que a sessão do jogo foi encerrada.
- `GAME_SESSION_ENDED_FAILED` – A sessão de jogo do Amazon GameLift Server não foi ativada.
- `GAME_SESSION_READY_FAILED` – O Server SDK do Amazon GameLift não conseguiu entrar em contato com o serviço para informar que a sessão do jogo está pronta.
- `INITIALIZATION_MISMATCH` – Um método de cliente foi chamado depois de `Server::Initialize()`, ou vice-versa.
- `NOT_INITIALIZED` – O servidor ou cliente do Amazon GameLift não foi inicializado com `Initialize()`.
- `NO_TARGET_ALIASID_SET` – Um `aliasID` de destino não foi definido.
- `NO_TARGET_FLEET_SET` – Uma frota alvo não foi definida.
- `PROCESS_ENDING_FAILED` – O Server SDK do Amazon GameLift não conseguiu entrar em contato com o serviço para informar que o processo está sendo encerrado.
- `PROCESS_NOT_ACTIVE` – O processo do servidor ainda não está ativo, não está vinculado a uma `GameSession` e não pode aceitar ou processar `PlayerSessions`.
- `PROCESS_NOT_READY` – O processo do servidor ainda não está pronto para ser ativado.

- `PROCESS_READY_FAILED` – O Server SDK do Amazon GameLift não conseguiu entrar em contato com o serviço para informar que o processo está pronto.
- `SDK_VERSION_DETECTION_FAILED` – Falha na detecção da versão do SDK.
- `STX_CALL_FAILED` – Uma chamada para o componente de back-end do servidor xSTx falhou.
- `STX_INITIALIZATION_FAILED` – O componente de back-end do servidor XSTx falhou ao inicializar.
- `UNEXPECTED_PLAYER_SESSION` – Uma sessão de jogador não registrada foi encontrada pelo servidor.
- `WEBSOCKET_CONNECT_FAILURE`
- `WEBSOCKET_CONNECT_FAILURE_FORBIDDEN`
- `WEBSOCKET_CONNECT_FAILURE_INVALID_URL`
- `WEBSOCKET_CONNECT_FAILURE_TIMEOUT`
- `WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE` – Falha recuperável ao enviar uma mensagem para o WebSocket do serviço GameLift.
- `WEBSOCKET_SEND_MESSAGE_FAILURE` – Falha ao enviar uma mensagem para o WebSocket do serviço GameLift.
- `MATCH_BACKFILL_REQUEST_VALIDATION` – A validação da solicitação falhou.
- `PLAYER_SESSION_REQUEST_VALIDATION` – A validação da solicitação falhou.

PlayerSessionCreationPolicy

`newPlayerSessionPolicy` Os valores válidos são:

- `ACCEPT_ALL` – Aceite todas as novas sessões de jogador.
- `DENY_ALL` – Recuse todas as novas sessões de jogador.
- `NOT_SET` – A sessão do jogo não está configurada para aceitar ou negar sessões de novos jogadores.

Status da sessão do jogador

- `ACTIVE`
- `COMPLETED`
- `NOT_SET`
- `RESERVED`
- `TIMEDOUT`

Referência 4.x do SDK do servidor Amazon GameLift para C#

Esta referência do Server SDK 4.x do Amazon GameLift C# pode ajudá-lo a preparar seu jogo multijogador a ser usado com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência do SDK do servidor do Amazon GameLift \(C#\): Ações](#)
- [Referência do SDK do servidor Amazon GameLift \(C#\): tipos de dados](#)

Referência do SDK do servidor do Amazon GameLift (C#): Ações

Você poderá usar essa referência do SDK do servidor Amazon GameLift C# para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

- [Ações](#)
- [Tipos de dados](#)

AcceptPlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se conectou ao processo do servidor e precisa de validação. O Amazon GameLift verifica se o ID da sessão do jogador é válido, ou seja, se o ID do jogador reservou um slot de jogador na sessão do jogo. Depois de validado, o Amazon GameLift altera o status do slot do jogador de RESERVED para ACTIVE.

Sintaxe

```
GenericOutcome AcceptPlayerSession(String playerSessionId)
```

Parâmetros

playerSessionId

ID exclusiva emitida pelo Amazon GameLift quando uma nova sessão de jogador é criada. O ID da sessão do jogador é especificado em um objeto `PlayerSession`, que

é retornado em resposta a uma chamada do cliente para as ações da API do GameLift [StartGameSessionPlacement](#), [CreateGameSession](#), [DescribeGameSessionPlacement](#) ou [DescribePlayerSessions](#).

Tipo: sequência

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra uma função para processar uma solicitação de conexão, inclusive validando e rejeitando IDs de sessão do jogador inválidos.

```
void ReceiveConnectingPlayerSessionID (Connection connection, String playerId){
    var acceptPlayerSessionOutcome =
    GameLiftServerAPI.AcceptPlayerSession(playerSessionId);
    if(acceptPlayerSessionOutcome.Success)
    {
        connectionToSessionMap.emplace(connection, playerId);
        connection.Accept();
    }
    else
    {
        connection.Reject(acceptPlayerSessionOutcome.Error.ErrorMessage);    }
}
```

ActivateGameSession()

Notifica o serviço do Amazon GameLift de que o processo do servidor ativou uma sessão do jogo e já está pronto para receber as conexões do jogador. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, depois que toda a inicialização da sessão tiver sido concluída.

Sintaxe

```
GenericOutcome ActivateGameSession()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo mostra `ActivateGameSession()` sendo chamado como parte da função de delegação `onStartGameSession()`.

```
void OnStartGameSession(GameSession gameSession)
{
    // game-specific tasks when starting a new game session, such as loading map

    // When ready to receive players
    var activateGameSessionOutcome = GameLiftServerAPI.ActivateGameSession();
}
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, inclusive configurações, metadados da sessão e dados do jogador. Use essa ação para obter informações de uma única sessão de jogador, para todas as sessões de jogador em uma sessão de jogo, ou para todas as sessões de jogador associadas a um único ID de jogador.

Sintaxe

```
DescribePlayerSessionsOutcome DescribePlayerSessions(DescribePlayerSessionsRequest
describePlayerSessionsRequest)
```

Parâmetros

describePlayerSessionsRequest

Um objeto [DescribePlayerSessionsRequest](#) que descreve quais sessões de jogador recuperar.

Obrigatório: Sim

Valor de retorno

Se bem-sucedido, retorna um objeto `DescribePlayerSessionsOutcome` que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação. Os objetos de sessão de jogador têm uma estrutura idêntica à do tipo de dados [PlayerSession](#) da API do SDKAWS do Amazon GameLift.

Exemplo

Este exemplo ilustra uma solicitação para todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Omitindo `NextToken` e definindo o valor `Limit` como 10, o Amazon GameLift retornará os 10 primeiros registros de sessões do jogador correspondentes à solicitação.

```
// Set request parameters
var describePlayerSessionsRequest = new
    Aws.GameLift.Server.Model.DescribePlayerSessionsRequest()
{
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result,    //gets the ID for
the current game session
    Limit = 10,
    PlayerSessionStatusFilter =
    PlayerSessionStatusMapper.GetNameForPlayerSessionStatus(PlayerSessionStatus.ACTIVE)
};
// Call DescribePlayerSessions
Aws::GameLift::DescribePlayerSessionsOutcome playerSessionsOutcome =
    Aws::GameLift::Server::Model::DescribePlayerSessions(describePlayerSessionRequest);
```

GetGameSessionId()

Recupera o ID da sessão do jogo hospedada no momento pelo processo do servidor, caso o processo do servidor esteja ativo.

Para processos inativos que ainda não foram ativados com uma sessão de jogo, a chamada retorna `Success=True` e `GameSessionId=""` (uma string vazia).

Sintaxe

```
AwsStringOutcome GetGameSessionId()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto `AwsStringOutcome`. Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
var getSessionIdOutcome = GameLiftServerAPI.GetSessionId();
```

GetInstanceCertificate()

Recupera a localização do arquivo de um certificado TLS codificado por pem que está associado à frota e suas instâncias. O AWS Certificate Manager gera esse certificado quando você cria uma nova frota com a configuração do certificado definida como `GENERATED`. Use esse certificado para estabelecer uma conexão segura com um cliente de jogo e para criptografar a comunicação entre cliente e servidor.

Sintaxe

```
GetInstanceCertificateOutcome GetInstanceCertificate();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se for bem-sucedido, o retorno será um objeto `GetInstanceCertificateOutcome` que contém a localização do arquivo de certificado TLS e da cadeia de certificados da frota, que estão armazenados na instância. Um arquivo de certificado raiz, extraído da cadeia de certificados, também é armazenado na instância. Se não for bem-sucedido, retornará uma mensagem de erro.

Para obter mais informações sobre o certificado e os dados da cadeia de certificados, consulte [Elementos de resposta GetCertificate](#) na Referência da API do AWS Certificate Manager.

Exemplo

```
var getInstanceCertificateOutcome = GameLiftServerAPI.GetInstanceCertificate();
```

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
AwsStringOutcome GetSdkVersion()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto `AwsStringOutcome`. A string retornada inclui apenas o número da versão (exemplo, "3.1.5"). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
var getSdkVersionOutcome = GameLiftServerAPI.GetSdkVersion();
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado, se essa informação estiver disponível. Um processo do servidor realiza essa ação depois de receber um retorno de chamada `onProcessTerminate()` do serviço Amazon GameLift. O Amazon GameLift pode chamar `onProcessTerminate()` pelos seguintes motivos: (1) por problemas de saúde (o processo do servidor relatou a integridade da porta ou não respondeu ao Amazon GameLift); (2) ao encerrar a instância durante um evento de redução ou (3) quando uma instância está sendo encerrada devido a uma [interrupção de instância spot](#).

Se o processo tiver recebido um retorno de chamada `onProcessTerminate()`, o valor retornado será o tempo estimado de encerramento. Se o processo não tiver recebido um retorno de chamada `onProcessTerminate()`, uma mensagem de erro será retornada. Saiba mais sobre [como desligar um processo do servidor](#).

Sintaxe

```
AwsDateTimeOutcome GetTerminationTime()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se for bem-sucedido, retornará o horário de término como um objeto `AwsDateTimeOutcome`. O valor é o tempo de término, expresso em tiques decorridos desde 0001 00:00:00. Por exemplo, o valor da data e hora 2020-09-13 12:26:40 -000Z é igual aos tiques 637355968000000000. Se nenhum horário de rescisão estiver disponível, o retornará uma mensagem de erro.

Exemplo

```
var getTerminationTimeOutcome = GameLiftServerAPI.GetTerminationTime();
```

InitSDK()

Inicializa o SDK do Amazon GameLift. Esse método deve ser chamado na inicialização antes de qualquer outra inicialização relacionada ao Amazon GameLift.

Sintaxe

```
InitSDKOutcome InitSDK()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará um objeto `InitSdkOutcome` indicando que o processo de servidor está pronto para chamar [ProcessReady\(\)](#).

Exemplo

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

ProcessEnding()

Notifica o serviço do Amazon GameLift de que o processo do servidor está sendo desligado. Esse método deverá ser chamado depois de todas as outras tarefas de limpeza, inclusive desligar todas

as sessões de jogos ativas. Este método deve sair com um código de saída zero; um código de saída diferente de zero resulta em uma mensagem de evento indicando que o processo não foi encerrado corretamente.

Depois que o método sair com um código de 0, você poderá encerrar o processo com um código de saída bem-sucedido. Você também poderá sair do processo com um código de erro. Se você sair com um código de erro, o evento da frota indicará que o processo foi encerrado de forma anormal (SERVER_PROCESS_TERMINATED_UNHEALTHY).

Sintaxe

```
GenericOutcome ProcessEnding()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
var processEndingOutcome = GameLiftServerAPI.ProcessEnding();
if (processReadyOutcome.Success)
    Environment.Exit(0);
// otherwise, exit with error code
Environment.Exit(errorCode);
```

ProcessReady()

Notifica o serviço Amazon GameLift de que o processo do servidor está pronto para hospedar sessões do jogo. Chame esse método depois de invocar [InitSDK\(\)](#) e concluir todas as tarefas de configuração necessárias antes que o processo do servidor possa hospedar uma sessão do jogo. Esse método deve ser chamado somente uma vez por processo.

Sintaxe

```
GenericOutcome ProcessReady(ProcessParameters processParameters)
```

Parâmetros

processParameters

Um objeto [ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes de métodos de retorno de chamada, implementados no código do servidor de jogos, que o serviço do Amazon GameLift invoca para se comunicar com o processo de servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho de qualquer arquivo específico da sessão do jogo que você deseja que o Amazon GameLift capture e armazene.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra as implementações das funções de chamada e delegação [ProcessReady\(\)](#).

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

var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);

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

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

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

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

RemovePlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se desconectou do processo do servidor. Em resposta, o Amazon GameLift altera o slot do jogador para um disponível, o que permite ser atribuído a um novo jogador.

Sintaxe

```
GenericOutcome RemovePlayerSession(String playerSessionId)
```

Parâmetros

playerSessionId

ID exclusiva emitida pelo Amazon GameLift quando uma nova sessão de jogador é criada. O ID da sessão do jogador é especificado em um objeto `PlayerSession`, que é retornado em resposta a uma chamada do cliente para as ações da API do GameLift [StartGameSessionPlacement](#), [CreateGameSession](#), [DescribeGameSessionPlacement](#) ou [DescribePlayerSessions](#).

Tipo: sequência

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
Aws::GameLift::GenericOutcome disconnectOutcome =  
    Aws::GameLift::Server::RemovePlayerSession(playerSessionId);
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Consulte também a ação [StopMatchmaking\(\)](#) do SDK da AWS. Com essa ação, as solicitações de alocação de correspondência podem ser iniciadas por um processo do servidor de jogos que esteja hospedando a sessão do jogo. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Esta ação é assíncrona. Se a correspondência dos novos jogadores for bem-sucedida, o serviço do Amazon GameLift entregará os dados atualizados do marcador de jogos usando a função de retorno de chamada `OnUpdateGameSession()`.

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
StartMatchBackfillOutcome StartMatchBackfill (StartMatchBackfillRequest  
    startBackfillRequest);
```

Parâmetros

StartMatchBackfillRequest

Um objeto [StartMatchBackfillRequest](#) que fornece as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essa informação é opcional. Caso nenhum ID seja fornecido, o Amazon GameLift gerará um automaticamente.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor pode ser obtido dos dados do marcador da sessão do jogo.
- ID da sessão de jogo que está sendo alocada.

- Dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Obrigatório: Sim

Valor de retorno

Retorna um objeto `StartMatchBackfillOutcome` com o ID do tíquete de alocação de correspondência ou um falha com uma mensagem de erro.

Exemplo

```
// Build a backfill request
var startBackfillRequest = new AWS.GameLift.Server.Model.StartMatchBackfillRequest()
{
    TicketId = "a ticket ID", //optional
    MatchmakingConfigurationArn = "the matchmaker configuration ARN",
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result, // gets ID for
current game session
    //get player data for all currently connected players
    MatchmakerData matchmakerData =
        MatchmakerData.FromJson(gameSession.MatchmakerData); // gets matchmaker
data for current players
    // get matchmakerData.Players
    // remove data for players who are no longer connected
    Players = ListOfPlayersRemainingInTheGame
};

// Send backfill request
var startBackfillOutcome = GameLiftServerAPI.StartMatchBackfill(startBackfillRequest);

// Implement callback function for backfill
void OnUpdateGameSession(GameSession myGameSession)
{
    // game-specific tasks to prepare for the newly matched players and update
matchmaker data as needed
}
```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa criada com [StartMatchBackfill\(\)](#). Consulte também a ação [StopMatchmaking\(\)](#) do SDK da AWS. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Sintaxe

```
GenericOutcome StopMatchBackfill (StopMatchBackfillRequest stopBackfillRequest);
```

Parâmetros

StopMatchBackfillRequest

Um objeto [StopMatchBackfillRequest](#) que identifica o tíquete de marcação de jogos para cancelar:

- ID do tíquete atribuído à solicitação de alocação sendo cancelada
- o marcador de jogo que recebeu a solicitação de alocação
- a sessão do jogo associada à solicitação de alocação

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
// Set backfill stop request parameters

var stopBackfillRequest = new AWS.GameLift.Server.Model.StopMatchBackfillRequest()
{
    TicketId = "a ticket ID", //optional, if not provided one is autogenerated
    MatchmakingConfigurationArn = "the matchmaker configuration ARN", //from the game
    session matchmaker data
    GameSessionId = GameLiftServerAPI.GetGameSessionId().Result //gets the ID for
    the current game session
};

var stopBackfillOutcome =
    GameLiftServerAPI.StopMatchBackfillRequest(stopBackfillRequest);
```

TerminateGameSession()

Esse método foi descontinuado com a versão 4.0.1. Em vez disso, o processo do servidor deve ser chamado [ProcessEnding\(\)](#) após o término de uma sessão de jogo.

Notifica o serviço do Amazon GameLift de que o processo do servidor encerrou a sessão atual do jogo. Essa ação é chamada quando o processo do servidor permanece ativo e pronto para hospedar uma nova sessão de jogo. Ele deve ser chamado somente após a conclusão do procedimento de encerramento da sessão de jogo, pois indica ao Amazon GameLift que o processo do servidor está imediatamente disponível para hospedar uma nova sessão de jogo.

Essa ação não será chamada se o processo do servidor for encerrado após o término da sessão do jogo. Em vez disso, chame [ProcessEnding\(\)](#) para sinalizar que tanto a sessão do jogo quanto o processo do servidor estão terminando.

Sintaxe

```
GenericOutcome TerminateGameSession()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra um processo de servidor no final de uma sessão de jogo.

```
// game-specific tasks required to gracefully shut down a game session,  
// such as notifying players, preserving game state data, and other cleanup  
  
var terminateGameSessionOutcome = GameLiftServerAPI.TerminateGameSession();  
var processReadyOutcome = GameLiftServerAPI.ProcessReady(processParams);
```

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. (Consulte também a ação [UpdateGameSession\(\)](#) na Referência de API do serviço do Amazon GameLift).

Sintaxe

```
GenericOutcome UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy  
playerSessionPolicy)
```

Parâmetros

newPlayerSessionPolicy

Valor de string que indica se a sessão do jogo aceita novos jogadores.

Tipo: enumeração [PlayerSessionCreationPolicy](#). Os valores válidos são:

- ACCEPT_ALL – Aceite todas as novas sessões de jogador.
- DENY_ALL – Recuse todas as novas sessões de jogador.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
var updatePlayerSessionCreationPolicyOutcome =  
    GameLiftServerAPI.UpdatePlayerSessionCreationPolicy(PlayerSessionCreationPolicy.ACCEPT_ALL);
```

Referência do SDK do servidor Amazon GameLift (C#): tipos de dados

Você poderá usar essa referência do SDK do servidor Amazon GameLift C# para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

- [Ações](#)
- Tipos de dados

LogParameters

Este tipo de dados é usado para identificar quais arquivos gerados durante uma sessão de jogo você deseja que o Amazon GameLift faça upload e armazene assim que a sessão de jogo terminar. Essas informações são comunicadas ao serviço do Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Índice

logPaths

Lista de caminhos de diretório para arquivos de log do servidor de jogo que você deseja que o Amazon GameLift armazene para acesso futuro. Esses arquivos são gerados por um processo de servidor durante cada sessão de jogo; caminhos de arquivo e nomes são definidos no servidor de jogo e armazenados no diretório de compilação do jogo raiz. Os caminhos de registro devem ser caminhos absolutos. Por exemplo, se a compilação do jogo armazenasse logs de sessão de jogo em um caminho como `MyGame\sessionlogs\`, o caminho de log seria `c:\game\MyGame\sessionLogs` (em uma instância do Windows) ou `/local/game/MyGame/sessionLogs` (em uma instância do Linux).

Tipo: `List<String>`

Obrigatório: não

DescribePlayerSessionsRequest

Esse tipo de dados é usado para especificar quais sessões de jogador recuperar. Ele pode ser usado de várias maneiras: (1) fornecer um `PlayerSessionId` para solicitar uma sessão de jogador específica; (2) fornecer um `GameSessionId` para solicitar todas as sessões de jogador na sessão de jogo especificada ou (3) fornecer um `PlayerId` para solicitar todas as sessões de jogador para o jogador especificado. Para coleções maiores de sessões de jogador, use os parâmetros de paginação para recuperar resultados como páginas sequenciais.

Índice

GameSessionId

Identificador da sessão de jogo exclusivo. Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada. O formato do ID da sessão de jogo é o seguinte: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. O valor de `<ID string>` é uma string de ID personalizada (se uma tiver sido especificada quando a sessão do jogo foi criada) ou uma string gerada.

Tipo: sequência

Obrigatório: não

Limite

O número máximo de resultados a serem retornados. Use esse parâmetro com NextToken para obter resultados como um conjunto de páginas sequenciais. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: inteiro

Obrigatório: não

NextToken

Token indicando o início da próxima página sequencial de resultados. Use o token retornado com uma chamada anterior para essa ação. Para especificar o início do conjunto de resultados, não especifique um valor. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: sequência

Obrigatório: não

PlayerId

Identificador exclusivo para um jogador. Os IDs de jogador são definidos pelo desenvolvedor. Consulte [Gerar IDs de jogador](#).

Tipo: sequência

Obrigatório: não

PlayerSessionId

Identificador exclusivo para uma sessão de jogador.

Tipo: sequência

Obrigatório: não

PlayerSessionStatusFilter

Status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis estão os seguintes:

- RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor e/ou foi validado.

- ACTIVE – O jogador foi validado pelo processo de servidor e está conectado no momento.
- COMPLETED – A conexão do jogador foi interrompida.
- TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou e/ou não foi validado dentro do tempo limite (60 segundos).

Tipo: sequência

Obrigatório: não

ProcessParameters

Este tipo de dados contém o conjunto de parâmetros enviado para o serviço do Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Índice

porta

Número da porta em que o processo de servidor escutará novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Tipo: inteiro

Obrigatório: Sim

logParameters

Objeto com uma lista de caminhos de diretório para arquivos de log da sessão de jogo.

Digite: `Aws::GameLift::Server::`[LogParameters](#)

Obrigatório: Sim

onStartGameSession

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para ativar uma nova sessão de jogo. O Amazon GameLift chama essa função em resposta à solicitação do cliente [CreateGameSession](#). A função de retorno de chamada utiliza um objeto [GameSession](#) (definido na Referência de API do serviço do Amazon GameLift).

Digite: `void OnStartGameSessionDelegate(GameSession gameSession)`

Obrigatório: Sim

onProcessTerminate

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para forçar o desligamento do processo de servidor. Depois de chamar essa função, o Amazon GameLift aguardará cinco minutos para o processo de servidor desligar e responder com uma chamada [ProcessEnding\(\)](#) antes do desligamento do processo de servidor.

Digite: `void OnProcessTerminateDelegate()`

Obrigatório: Sim

onHealthCheck

Nome da função de retorno que o serviço do Amazon GameLift invoca para solicitar um relatório de status de integridade do processo de servidor. O Amazon GameLift chama essa função a cada 60 segundos. Depois de chamar essa função, o Amazon GameLift aguardará 60 segundos por uma resposta e, se nenhuma for recebida, registra o processo do servidor como não íntegro.

Digite: `bool OnHealthCheckDelegate()`

Obrigatório: Sim

onUpdateGameSession

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. O Amazon GameLift chama essa função quando uma solicitação de [alocação de correspondência](#) é processada para fornecer dados atualizados do marcador de jogos. Ele passa um objeto [GameSession](#), uma atualização de status (`updateReason`) e o ID do tíquete de alocação de correspondência.

Digite: `void OnUpdateGameSessionDelegate (UpdateGameSession updateGameSession)`

Obrigatório: não

StartMatchBackfillRequest

Esse tipo de dados é usado para enviar uma solicitação de alocação de marcação de jogos. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StartMatchBackfill\(\)](#).

Índice

GameSessionArn

Identificador da sessão de jogo exclusivo. O método [GetGameSessionId\(\)](#) do SDK retorna o identificador no formato de ARN.

Tipo: sequência

Obrigatório: Sim

MatchmakingConfigurationArn

O identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. Para encontrar o marcador de jogos usado para criar a sessão de jogo original, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Saiba mais sobre os dados de marcador de jogos em [Trabalhar com dados do marcador de jogos](#).

Tipo: sequência

Obrigatório: Sim

Jogadores

Um conjunto de dados que representa todos os jogadores que estão na sessão do jogo no momento. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais. Consulte o Guia de Referência de API do Amazon GameLift para obter uma descrição do formato do objeto Player. Para encontrar atributos, IDs e atribuições de equipe dos jogadores, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Se o marcador de jogos usa latência, obtenha a latência atualizada para a região atual e a inclua nos dados de cada jogador.

Tipo: [Player](#)[]

Obrigatório: Sim

TicketId

O identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se nenhum valor for fornecido aqui, o Amazon GameLift gerará um no formato de UUID. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.

Tipo: sequência

Obrigatório: não

StopMatchBackfillRequest

Esse tipo de dados é usado para cancelar uma solicitação de alocação de marcação de jogo. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StopMatchBackfill\(\)](#).

Índice

GameSessionArn

O identificador de sessão de jogo exclusivo associado à solicitação que está sendo cancelada.

Tipo: sequência

Obrigatório: Sim

MatchmakingConfigurationArn

O identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada.

Tipo: sequência

Obrigatório: Sim

TicketId

O identificador exclusivo do tíquete de solicitação de alocação a ser cancelado.

Tipo: sequência

Obrigatório: Sim

Referência do SDK do servidor do Amazon GameLift para Go

Você poderá usar essa referência do SDK de servidor do Amazon GameLift para Go para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência GameLift do SDK \(Go\) do servidor Amazon: Ações](#)
- [Referência GameLift do SDK \(Go\) do servidor Amazon: tipos de dados](#)

Referência GameLift do SDK (Go) do servidor Amazon: Ações

Você pode usar essa referência do SDK do servidor Amazon GameLift Go para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

GameLiftServerAPI.go define as ações do SDK do servidor Go.

Ações

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetTerminationTime\(\)](#)
- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)
- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)
- [Destroy\(\)](#)

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
func GetSdkVersion() (string, error)
```

Valor de retorno

Se bem-sucedido, retornará a versão do SDK como uma sequência. A string retornada inclui o número da versão (exemplo, 5.0.0). Se não for bem-sucedido, retornará uma mensagem de erro como `common.SdkVersionDetectionFailed`.

Exemplo

```
version, err := server.GetSdkVersion()
```

InitSDK()

Inicializa o Amazon GameLift SDK. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método configura a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
func InitSDK(params ServerParameters) error
```

Parâmetros

[ServerParameters](#)

Para inicializar um servidor de jogo em uma GameLift Anywhere frota da Amazon, construa um `ServerParameters` objeto com as seguintes informações:

- O URL do WebSocket usado para se conectar ao seu servidor de jogo.
- O ID do processo usado para hospedar o servidor de jogos.
- O ID do computador que hospeda os processos do seu servidor de jogos.
- O ID da GameLift frota da Amazon que contém sua GameLift Anywhere computação da Amazon.
- O token de autorização gerado pela GameLift operação da Amazon.

Para inicializar um servidor de jogo em uma frota EC2 GameLift gerenciada pela Amazon, construa um `ServerParameters` objeto sem parâmetros. Com essa chamada, o GameLift agente da Amazon configura o ambiente computacional e se conecta automaticamente ao GameLift serviço da Amazon para você.

Valor de retorno

Se for bem-sucedido, retornará um erro `nil` para indicar que o processo do servidor está pronto para ser chamado [ProcessReady\(\)](#).

Note

Se as chamadas para `InitSDK()` estiverem falhando para compilações de jogos implantadas em frotas do Anywhere, verifique o parâmetro `ServerSdkVersion` usado ao criar o recurso de compilação. Você deve definir explicitamente esse valor para a versão do SDK do servidor em uso. O valor padrão desse parâmetro é 4.x, o que não é compatível. Para resolver esse problema, crie uma nova versão e implante-a em uma nova frota.

Exemplo

GameLift Anywhere Exemplo da Amazon

```
//Define the server parameters
serverParameters := ServerParameters {
  WebSocketURL: "wss://us-west-1.api.amazongamelift.com",
  ProcessID: "PID1234",
  HostID: "HardwareAnywhere",
  FleetID: "aarn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa",
  AuthToken: "1111aaaa-22bb-33cc-44dd-5555eeee66ff"
}

//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
err := server.InitSDK(serverParameters)
```

Exemplo de EC2 GameLift gerenciado pela Amazon

```
//Define the server parameters
serverParameters := ServerParameters {}

//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
err := server.InitSDK(serverParameters)
```

ProcessReady()

Notifica a Amazon de GameLift que o processo do servidor está pronto para hospedar sessões de jogo. Chame esse método após invocar [InitSDK\(\)](#). Esse método deve ser chamado apenas uma vez por processo.

Sintaxe

```
func ProcessReady(param ProcessParameters) error
```

Parâmetros

ProcessParameters

Um objeto [ProcessParameters](#) transmite as seguintes informações sobre o processo do servidor:

- Os nomes dos métodos de retorno de chamada implementados no código do servidor do jogo que o GameLift serviço da Amazon invoca para se comunicar com o processo do servidor.
- O número da porta em que o processo de servidor está escutando.
- O tipo de [LogParameters](#) dados que contém o caminho para qualquer arquivo específico da sessão de jogo que você deseja que GameLift a Amazon capture e armazene.

Valor de retorno

Retorna um erro com uma mensagem de erro se o método falhar. Se o método for bem-sucedido, ele retornará `nil`.

Exemplo

Este exemplo ilustra as implementações das funções de chamada e delegação [ProcessReady\(\)](#).

```
// Define the process parameters
processParams := ProcessParameters {
    OnStartGameSession: gameProcess.OnStartGameSession,
    OnUpdateGameSession: gameProcess.OnGameSessionUpdate,
    OnProcessTerminate: gameProcess.OnProcessTerminate,
    OnHealthCheck: gameProcess.OnHealthCheck,
    Port: port,
    LogParameters: LogParameters { // logging and error example
        []string {"C:\\game\\logs", "C:\\game\\error"}
```

```
    }  
  }  
  
  err := server.ProcessReady(processParams)
```

ProcessEnding()

Notifica a Amazon de GameLift que o processo do servidor está sendo encerrado. Chame esse método depois de todas as outras tarefas de limpeza (incluindo o encerramento da sessão ativa do jogo) e antes de encerrar o processo. Dependendo do resultado `ProcessEnding()`, o processo sai com êxito (0) ou erro (-1) e gera um evento de frota. Se o processo for encerrado com um erro, o evento de frota gerado será `SERVER_PROCESS_TERMINATED_UNHEALTHY`.

Sintaxe

```
func ProcessEnding() error
```

Valor de retorno

Retorna um código de erro 0 ou um código de erro definido.

Exemplo

```
// operations to end game sessions and the server process  
defer func() {  
    err := server.ProcessEnding()  
    server.Destroy()  
    if err != nil {  
        fmt.Println("ProcessEnding() failed. Error: ", err)  
        os.Exit(-1)  
    } else {  
        os.Exit(0)  
    }  
}
```

ActivateGameSession()

Notifica a Amazon de GameLift que o processo do servidor ativou uma sessão de jogo e agora está pronto para receber conexões de jogadores. Essa ação é chamada como parte da função de retorno de chamada `onStartGameSession()`, após toda a inicialização da sessão do jogo.

Sintaxe

```
func ActivateGameSession() error
```

Valor de retorno

Retorna um erro com uma mensagem de erro se o método falhar.

Exemplo

Este exemplo mostra o chamado `ActivateGameSession()` como parte da função de delegação `onStartGameSession()`.

```
func OnStartGameSession(GameSession gameSession) {  
    // game-specific tasks when starting a new game session, such as loading map  
    // Activate when ready to receive players  
    err := server.ActivateGameSession();  
}
```

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador.

Sintaxe

```
func UpdatePlayerSessionCreationPolicy(policy model.PlayerSessionCreationPolicy) error
```

Parâmetros

playerSessionCreationPolítica

Valor de string que indica se a sessão do jogo aceita novos jogadores.

Os valores válidos são:

- **model.AcceptAll** – Aceite todas as novas sessões de jogador.
- **model.DenyAll** – Recuse todas as novas sessões de jogador.

Valor de retorno

Retorna um erro com uma mensagem de erro se ocorrer uma falha.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
err := server.UpdatePlayerSessionCreationPolicy(model.AcceptAll)
```

GetGameSessionId()

Recupera o ID da sessão de jogo hospedada pelo processo do servidor ativo.

Sintaxe

```
func GetGameSessionID() (string, error)
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo e nenhum erro. Para processos inativos que ainda não foram ativados com uma sessão de jogo, a chamada retorna uma string vazia e um erro `nil`.

Exemplo

```
gameSessionID, err := server.GetGameSessionID()
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado se essa informação estiver disponível. Um processo de servidor executa essa ação após receber um `onProcessTerminate()` retorno de chamada da Amazon GameLift. A Amazon GameLift liga `onProcessTerminate()` pelos seguintes motivos:

- Quando o processo do servidor relatou problemas de saúde ou não respondeu à Amazon GameLift.
- Ao encerrar a instância durante um evento de redução.
- Quando uma instância é encerrada devido a uma [interrupção na instância spot](#).

Sintaxe

```
func GetTerminationTime() (int64, error)
```

Valor de retorno

Se for bem-sucedido, retornará o registro de data e hora em segundos em que o processo do servidor está programado para ser encerrado e um erro nil de encerramento. O valor é o tempo de rescisão, expresso em tiques decorridos de 0001 00:00:00. Por exemplo, o valor da data e hora 2020-09-13 12:26:40 -000Z é igual aos tiques 637355968000000000. Se nenhum horário de rescisão estiver disponível, o retornará uma mensagem de erro.

Exemplo

```
terminationTime, err := server.GetTerminationTime()
```

AcceptPlayerSession()

Notifica a Amazon de GameLift que um jogador com o ID de sessão de jogador especificado se conectou ao processo do servidor e precisa de validação. A Amazon GameLift verifica se o ID da sessão do jogador é válido. Depois que a sessão do jogador é validada, a Amazon GameLift altera o status do slot do jogador de RESERVED para ACTIVE.

Sintaxe

```
func AcceptPlayerSession(playerSessionID string) error
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo trata de uma solicitação de conexão que inclui a validação e a rejeição de IDs de sessão de jogadores inválidas.

```
func ReceiveConnectingPlayerSessionID(conn Connection, playerSessionID string) {
    err := server.AcceptPlayerSession(playerSessionID)
    if err != nil {
        connection.Accept()
    } else {
        connection.Reject(err.Error())
    }
}
```

RemovePlayerSession()

Notifica a Amazon de GameLift que um jogador se desconectou do processo do servidor. Em resposta, a Amazon GameLift altera o slot do jogador para disponível.

Sintaxe

```
func RemovePlayerSession(playerSessionID string) error
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
err := server.RemovePlayerSession(playerSessionID)
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, que incluem configurações, metadados da sessão e dados do jogador. Use esse método para obter informações sobre o seguinte:

- Uma sessão para um jogador
- Todas as sessões de jogador em uma sessão de jogo
- Todas as sessões de jogadores associadas a um único ID de jogador

Sintaxe

```
func DescribePlayerSessions(req request.DescribePlayerSessionsRequest)
    (result.DescribePlayerSessionsResult, error) {
    return srv.describePlayerSessions(&req)
}
```

Parâmetros

[DescribePlayerSessionsRequest](#)

Um objeto `DescribePlayerSessionsRequest` descreve quais sessões de jogador recuperar.

Valor de retorno

Se bem-sucedido, retorna um objeto `DescribePlayerSessionsResult` que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação.

Exemplo

Este exemplo solicita todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Ao omitir `NextToken` definir o valor limite como 10, a Amazon GameLift retorna os primeiros registros de sessão de 10 jogadores correspondentes à solicitação.

```
// create request
describePlayerSessionsRequest := request.NewDescribePlayerSessions()
describePlayerSessionsRequest.GameSessionID, _ = server.GetGameSessionID() // get ID
for the current game session
describePlayerSessionsRequest.Limit = 10 // return the
first 10 player sessions
describePlayerSessionsRequest.PlayerSessionStatusFilter = "ACTIVE" // Get all
player sessions actively connected to the game session

describePlayerSessionsResult, err :=
server.DescribePlayerSessions(describePlayerSessionsRequest)
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Esta ação é assíncrona. Se novos jogadores forem combinados, a Amazon GameLift fornece dados atualizados do matchmaker usando a função de retorno de chamada. `OnUpdateGameSession()`

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
func StartMatchBackfill(req request.StartMatchBackfillRequest)
    (result.StartMatchBackfillResult, error)
```

Parâmetros

[StartMatchBackfillRequest](#)

Um `StartMatchBackfillRequest` objeto comunica as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essas informações são opcionais; se nenhum ID for fornecido, a Amazon GameLift gerará um.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor está nos dados do marcador de jogos da sessão de jogo.
- O ID da sessão de jogo a ser preenchida.
- Os dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Valor de retorno

Retorna um objeto `StartMatchBackfillResult` com o ID do tíquete de alocação de correspondência ou um falha com uma mensagem de erro.

Exemplo

```
// form the request
startBackfillRequest := request.NewStartMatchBackfill()
startBackfillRequest.RequestID = "1111aaaa-22bb-33cc-44dd-5555eeee66ff" //
    optional
startBackfillRequest.MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"
var matchMaker model.MatchmakerData
if err := matchMaker.UnmarshalJSON([]byte(gameSession.MatchmakerData)); err != nil {
```

```
    return
}
startBackfillRequest.Players = matchMaker.Players
res, err := server.StartMatchBackfill(startBackfillRequest)

// Implement callback function for backfill
func OnUpdateGameSession(myGameSession model.GameSession) {
    // game-specific tasks to prepare for the newly matched players and update
    matchmaker data as needed
}
```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Sintaxe

```
func StopMatchBackfill(req request.StopMatchBackfillRequest) error
```

Parâmetros

[StopMatchBackfillRequest](#)

Um `StopMatchBackfillRequest` objeto que identifica o tíquete de matchmaking a ser cancelado:

- O ID do ticket atribuído à solicitação de alocação.
- O marcador de jogo que recebeu a solicitação de alocação.
- A sessão do jogo associada à solicitação de alocação.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
stopBackfillRequest := request.NewStopMatchBackfill() // Use this function to create
request
stopBackfillRequest.TicketID = "1111aaaa-22bb-33cc-44dd-5555eeee66ff"
stopBackfillRequest.MatchmakingConfigurationArn = "arn:aws:gamelift:us-
west-2:111122223333:matchmakingconfiguration/MyMatchmakerConfig"
```

```
//error
err := server.StopMatchBackfill(stopBackfillRequest)
```

GetComputeCertificate()

Recupera o caminho para o certificado TLS usado para criptografar a conexão de rede entre o servidor do jogo e o cliente do jogo. Você pode usar o caminho do certificado ao registrar seu dispositivo computacional em uma GameLift Anywhere frota da Amazon. Para obter mais informações, consulte [RegisterCompute](#).

Sintaxe

```
func GetComputeCertificate() (result.GetComputeCertificateResult, error)
```

Valor de retorno

Retorna um objeto `GetComputeCertificateResult` que contém o seguinte:

- `CertificatePath`: o caminho para o certificado TLS em seu recurso computacional. Ao usar uma frota GameLift gerenciada pela Amazon, esse caminho contém:
 - `certificate.pem`: o certificado do usuário final. A cadeia completa de certificados é a combinação de `certificateChain.pem` anexados a esse certificado.
 - `certificateChain.pem`: a cadeia de certificados que contém o certificado raiz e os certificados intermediários.
 - `rootCertificate.pem`: Certificado raiz.
 - `privateKey.pem`: a chave privada para o certificado do usuário final.
- `ComputeName`: o nome do seu recurso computacional.

Exemplo

```
tlsCertificate, err := server.GetFleetRoleCredentials(getFleetRoleCredentialsRequest)
```

GetFleetRoleCredentials()

Recupera as credenciais da função de serviço que você cria para estender as permissões à sua outra pessoa Serviços da AWS na Amazon. GameLift Essas credenciais permitem que seu servidor

de jogo use seus recursos AWS. Para ter mais informações, consulte [Configurar uma função de serviço do IAM para a Amazon GameLift](#).

Sintaxe

```
func GetFleetRoleCredentials(  
    req request.GetFleetRoleCredentialsRequest,  
) (result.GetFleetRoleCredentialsResult, error) {  
    return srv.getFleetRoleCredentials(&req)  
}
```

Parâmetros

[GetFleetRoleCredentialsRequest](#)

Credenciais de função que ampliam o acesso limitado aos seus recursos AWS no servidor do jogo.

Valor de retorno

Retorna um objeto `GetFleetRoleCredentialsResult` que contém o seguinte:

- `AssumedRoleUserArn` - O Amazon Resource Name (ARN) do usuário ao qual a função de serviço pertence.
- `AssumedRoleId` - O ID do usuário ao qual a função de serviço pertence.
- `AccessKeyId` - O ID da chave de acesso para autenticar e fornecer acesso aos seus AWS recursos.
- `SecretAccessKey` - O ID da chave de acesso secreta para autenticação.
- `SessionToken` - Um token para identificar a sessão ativa atual interagindo com seus AWS recursos.
- `Expiração` – A quantidade de tempo até que suas credenciais de sessão expirem.

Exemplo

```
// form the customer credentials request  
getFleetRoleCredentialsRequest := request.NewGetFleetRoleCredentials()  
getFleetRoleCredentialsRequest.RoleArn = "arn:aws:iam::123456789012:role/service-role/  
exampleGameLiftAction"
```



```
credentials, err := server.GetFleetRoleCredentials(getFleetRoleCredentialsRequest)
```

Destroy()

Libera o SDK do servidor de GameLift jogos da Amazon da memória. Como prática recomendada, chame esse método antes `ProcessEnding()` e depois de encerrar o processo. Se você estiver usando uma frota Anywhere e não estiver encerrando os processos do servidor após cada sessão de jogo, ligue `Destroy()` e reinicialize antes de notificar `InitSDK()` a Amazon de GameLift que o processo está pronto para hospedar uma sessão de jogo. `ProcessReady()`

Sintaxe

```
func Destroy() error {  
    return srv.destroy()  
}
```

Valor de retorno

Retorna um erro com uma mensagem de erro se o método falhar.

Exemplo

```
// operations to end game sessions and the server process  
defer func() {  
    err := server.ProcessEnding()  
    server.Destroy()  
    if err != nil {  
        fmt.Println("ProcessEnding() failed. Error: ", err)  
        os.Exit(-1)  
    } else {  
        os.Exit(0)  
    }  
}
```

Referência GameLift do SDK (Go) do servidor Amazon: tipos de dados

Você pode usar essa referência do SDK do servidor Amazon GameLift Go para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tipos de dados

- [LogParameters](#)
- [ProcessParameters](#)
- [UpdateGameSession](#)
- [GameSession](#)
- [ServerParameters](#)
- [StartMatchBackfillRequest](#)
- [Jogador](#)
- [DescribePlayerSessionsRequest](#)
- [StopMatchBackfillRequest](#)
- [GetFleetRoleCredentialsRequest](#)

LogParameters

Um objeto que identifica arquivos gerados durante uma sessão de jogo que você deseja que GameLift a Amazon envie e armazene após o término da sessão de jogo. O servidor do jogo fornece `LogParameters` à Amazon GameLift como parte de um `ProcessParameters` objeto em uma [ProcessReady\(\)](#) chamada.

Properties	Descrição
LogPaths	<p>A lista de caminhos de diretório para os arquivos de log do servidor de jogos que você deseja que GameLift a Amazon armazene para acesso futuro. O processo do servidor gera esses arquivos durante cada sessão de jogo. Você define caminhos e nomes de arquivos no servidor de jogo e os armazena no diretório raiz de compilação do jogo.</p> <p>Os caminhos de log devem ser absolutos. Por exemplo, se a compilação do jogo armazenasse logs de sessão de jogo em um caminho como <code>MyGame\sessionLogs\</code>, o caminho estará em <code>c:\game\MyGame\sessionLogs</code> uma instância do Windows.</p> <p>Digite: <code>[]string</code></p> <p>Obrigatório: não</p>

ProcessParameters

Um objeto que descreve a comunicação entre um processo de servidor e a Amazon GameLift. O processo do servidor fornece essas informações para a Amazon GameLift com uma chamada para [ProcessReady\(\)](#).

Properties	Descrição
LogParameters	<p>Um objeto com caminhos de diretório para arquivos que são gerados durante uma sessão de jogo. A Amazon GameLift copia e armazena os arquivos para acesso futuro.</p> <p>Digite: LogParameters</p> <p>Obrigatório: não</p>
OnHealthCheck	<p>A função de retorno de chamada que a Amazon GameLift invoca para solicitar um relatório de status de saúde do processo do servidor. A Amazon GameLift chama essa função a cada 60 segundos e espera 60 segundos por uma resposta. O processo do servidor retorna TRUE se estiver íntegro e FALSE se não estiver íntegro. Se nenhuma resposta for retornada, a Amazon GameLift registra o processo do servidor como não íntegro.</p> <p>Digite: <code>OnHealthCheck func() bool</code></p> <p>Obrigatório: não</p>
OnProcessTerminate	<p>A função de retorno de chamada que a Amazon GameLift invoca para forçar o encerramento do processo do servidor. Depois de chamar essa função, a Amazon GameLift espera 5 minutos até que o processo do servidor seja encerrado e responda com uma ProcessEnding() chamada antes de encerrar o processo do servidor.</p> <p>Digite: <code>OnProcessTerminate func()</code></p> <p>Obrigatório: Sim</p>
OnStartGameSession	<p>A função de retorno de chamada que a Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. A Amazon GameLift chama essa função quando uma solicitação de preenchim</p>

ento de correspondência é processada para fornecer dados atualizados do matchmaker. Ele passa um [GameSession](#) objeto, uma atualização de status (updateReason) e o ID do tíquete de preenchimento correspondente.

Digite: `OnStartGameSession func (model.GameSession)`

Obrigatório: Sim

`OnUpdateGameSession`

A função de retorno de chamada que a Amazon GameLift invoca para passar informações atualizadas da sessão do jogo para o processo do servidor. A Amazon GameLift chama essa função após processar uma solicitação de preenchimento de correspondência para fornecer dados atualizados do matchmaker.

Digite: `OnUpdateGameSession func (model.UpdateGameSession)`

Obrigatório: não

`Port`

O número da porta em que o processo de servidor escuta para novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Digite: `int`

Obrigatório: Sim

UpdateGameSession

As atualizações em um objeto de sessão de jogo, que incluem o motivo pelo qual a sessão de jogo foi atualizada e o ID do tíquete de alocação relacionado, caso o preenchimento esteja sendo usado para preencher sessões de jogadores na sessão de jogo.

Propriedades	Descrição
GameSession	Um GameSession objeto definido pela GameLift API da Amazon. O objeto <code>GameSession</code> contém propriedades que descrevem uma sessão de jogo. Digite: <code>GameSession GameSession()</code> Obrigatório: Sim
UpdateReason	O motivo pelo qual a sessão do jogo está sendo atualizada. Digite: <code>UpdateReason UpdateReason()</code> Obrigatório: Sim
BackfillTicketId	O ID do tíquete de alocação que está tentando atualizar a sessão do jogo. Digite: <code>String</code> Obrigatório: não

GameSession

Os detalhes de uma sessão de jogo.

Propriedades	Descrição
GameSessionId	Um identificador exclusivo da sessão de jogo. Um nome do recurso da Amazon (ARN) de sessão de jogo tem o seguinte formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> . Digite: <code>String</code> Obrigatório: não
Nome	Um rótulo descritivo da sessão do jogo. Digite: <code>String</code>

Propriedades	Descrição
	Obrigatório: não
FleetId	Um identificador exclusivo da frota em que a sessão de jogo está sendo executada. Digite: <code>String</code> Obrigatório: não
MaximumPlayerSessionCount	O número máximo de conexões do jogador com a sessão do jogo. Digite: <code>Integer</code> Obrigatório: não
Porta	O número da porta para a sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta. Digite: <code>Integer</code> Obrigatório: não
IpAddress	O endereço IP da sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta. Digite: <code>String</code> Obrigatório: não
GameSessionData	Um conjunto de propriedades de sessão de jogo personalizadas, formatadas como um único valor de string. Digite: <code>String</code> Obrigatório: não

Propriedades	Descrição
MatchmakerData	<p>As informações sobre o processo de matchmaking usado para criar a sessão do jogo, na sintaxe JSON, formatado como uma string. Além da configuração de matchmaking usada, ele contém dados sobre todos os jogadores atribuídos à partida, incluindo atributos do jogador e atribuições da equipe.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
GameProperties	<p>Um conjunto de propriedades personalizadas para uma sessão de jogo, formatado como pares de chave:valor. Essas propriedades são passadas com uma solicitação de início de uma nova sessão de jogo.</p> <p>Digite: <code>map[string] string</code></p> <p>Obrigatório: não</p>
DnsName	<p>O identificador DNS atribuído à instância que está executando a sessão do jogo. Os valores têm o formato a seguir:</p> <ul style="list-style-type: none"> Frotas habilitadas para TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> Frotas não habilitadas para TLS: <code>ec2-<unique identifier>.compute.amazonaws.com</code> <p>Ao se conectar a uma sessão de jogo que está sendo executada em uma frota compatível com TLS, você deve usar o nome DNS, não o endereço IP.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

ServerParameters

Informações usadas para manter a conexão entre um GameLift Anywhere servidor da Amazon e o GameLift serviço da Amazon. Essas informações são usadas ao iniciar novos processos de servidor

com [InitSDK\(\)](#). Para servidores hospedados em instâncias EC2 GameLift gerenciadas pela Amazon, use um objeto vazio.

Propriedades	Descrição
WebSocket URL	<p>A <code>GameLiftServerSdkEndpoint</code> Amazon GameLift retorna quando você compra RegisterCompute um recurso GameLift Anywhere computacional da Amazon.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: Sim</p>
ProcessID	<p>Um identificador exclusivo registrado no processo do servidor que hospeda seu jogo.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: Sim</p>
HostID	<p>O identificador exclusivo do recurso de computação que está hospedando o novo processo de servidor.</p> <p>O <code>HostID</code> é o <code>ComputeName</code> usado quando você registrou seu computador. Para obter mais informações, consulte RegisterCompute.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: Sim</p>
FleetID	<p>O identificador exclusivo da frota na qual o computador está registrado. Para obter mais informações, consulte RegisterCompute.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: Sim</p>
AuthToken	<p>O token de autenticação gerado pela Amazon GameLift que autentica seu servidor na Amazon GameLift. Para obter mais informações, consulte GetComputeAuthToken.</p>

Propriedades	Descrição
	Digite: <code>string</code>
	Obrigatório: Sim

StartMatchBackfillRequest

Informações usadas para criar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações à Amazon GameLift em uma [StartMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	O identificador da sessão de jogo exclusivo. A operação de API retorna GetGameSessionId o identificador no formato de ARN. Digite: <code>String</code> Obrigatório: Sim
MatchmakingConfigurationArn	O identificador exclusivo (na forma de um ARN), para o marcador de jogos a ser usado para essa solicitação. O marcador de jogos da ARN da sessão de jogo original está no objeto de sessão de jogo, na propriedade de dados do marcador de jogos. Para obter mais informações sobre dados de matchmaker, consulte Trabalhar com dados de marcador de jogos . Digite: <code>String</code> Obrigatório: Sim
Jogadores	Um conjunto de dados que representa todos os jogadores que estão na sessão do jogo no momento. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais. Digite: <code>[]model.Player</code> Obrigatório: Sim

Propriedades	Descrição
TicketId	<p>O identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se você não fornecer um valor, a Amazon GameLift gera um. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

Jogador

O objeto que representa um jogador em marcação de jogos. Quando uma solicitação de marcação de jogos começa, um jogador tem um ID de jogador, atributos, e possivelmente dados de latência. A Amazon GameLift adiciona informações da equipe após a realização de uma partida.

Propriedades	Descrição
LatencyInMS	<p>Um conjunto de valores expressos em milissegundos que indicam a quantidade de latência que um jogador experimenta quando conectado a um local.</p> <p>Se essa propriedade for usada, o jogador só será correspondido aos locais listados. Se um marcador de jogos tiver uma regra que avalie a latência do jogador, os jogadores deverão reportar a latência a ser correspondida.</p> <p>Digite: <code>map[string] int</code></p> <p>Obrigatório: não</p>
PlayerAttributes	<p>Uma coleção de pares de chave:valor que contêm informações de jogadores para uso em marcação de jogos. As chaves de atributos do jogador devem corresponder às <code>PlayerAttributes</code> usadas em um conjunto de regras de matchmaking.</p> <p>Para obter mais informações sobre os atributos do jogador, consulte Attribute Value.</p>

Propriedades	Descrição
	Digite: <code>map[string] AttributeValue</code> Obrigatório: não
PlayerId	Um identificador exclusivo para um jogador. Digite: <code>String</code> Obrigatório: não
Team	O nome da equipe à qual o jogador está designado em uma partida. Você define o nome da equipe no conjunto de regras de marcação de jogos. Digite: <code>String</code> Obrigatório: não

DescribePlayerSessionsRequest

Um objeto que especifica quais sessões de jogador devem ser recuperadas. O processo do servidor fornece essas informações com uma [DescribePlayerSessions\(\)](#) chamada para a Amazon GameLift.

Propriedades	Descrição
GameSessionID	Um identificador da sessão de jogo exclusivo. Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada. O formato do ID da sessão do jogo é <code>arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string></code> . O <code>GameSessionID</code> é uma string de ID personalizada ou uma string gerada. Digite: <code>String</code> Obrigatório: não
PlayerSessionID	Um identificador exclusivo para uma sessão de jogador. Use esse parâmetro para solicitar uma única sessão de jogador específica. Digite: <code>String</code>

Propriedades	Descrição
	Obrigatório: não
PlayerID	<p>Um identificador exclusivo para um jogador. Use este parâmetro para solicitar todas as sessões de jogador para um jogador específico. Consulte Gerar IDs de jogador.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
PlayerSessionStatusFilter	<p>O status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis incluem:</p> <ul style="list-style-type: none">• RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor nem foi validado.• ACTIVE – O player foi validado pelo processo do servidor e está conectado.• COMPLETED – A conexão do jogador foi interrompida.• TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou ou não foi validado dentro do tempo limite (60 segundos). <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>
NextToken	<p>O token que indica o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>String</code></p> <p>Obrigatório: não</p>

Propriedades	Descrição
Limit	<p>O número máximo de resultados a serem retornados. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>

StopMatchBackfillRequest

Informações usadas para cancelar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações ao GameLift serviço da Amazon em uma [StopMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	<p>O identificador de sessão de jogo exclusivo da solicitação que está sendo cancelada.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
MatchmakingConfigurationArn	<p>O identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>
TicketId	<p>O identificador exclusivo do tíquete de solicitação de alocação a ser cancelado.</p> <p>Digite: <code>string</code></p> <p>Obrigatório: não</p>

GetFleetRoleCredentialsRequest

As credenciais da função que estendem o acesso limitado aos seus recursos AWS no servidor do jogo. Para obter mais informações, consulte, [Configurar uma função de serviço do IAM para a Amazon GameLift](#).

Propriedades	Descrição
RoleArn	O ARN do perfil de serviço que estende o acesso limitado aos seus recursos AWS. Digite: <code>string</code> Obrigatório: Sim
RoleSessionName	O nome da sessão que descreve o uso das credenciais da função. Digite: <code>string</code> Obrigatório: Sim

Referência do SDK do servidor do Amazon GameLift para Unreal Engine

Esta referência de Server SDK do Amazon GameLift pode ajudar a preparar os projetos de jogo Unreal Engine a serem usados com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Esta API é definida em `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Para configurar o plug-in do Unreal Engine e ver exemplos de código [Integre a Amazon GameLift em um projeto Unreal Engine](#).

Tópicos

- [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)
- [Referência do SDK 3.x do servidor do Amazon GameLift para Unreal Engine](#)

Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine

Você poderá usar essa referência SDK 5.x do servidor do Amazon GameLift para Unreal Engine para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in do SDK do servidor do Unreal, consulte [Integre a Amazon GameLift em um projeto Unreal Engine](#).

Tópicos

- [Referência 5.x GameLift do Amazon Server SDK \(Unreal\): Ações](#)
- [Referência GameLift do Amazon Server SDK \(Unreal\): tipos de dados](#)

Referência 5.x GameLift do Amazon Server SDK (Unreal): Ações

Você pode usar essa referência do SDK do servidor Amazon GameLift Unreal para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon. GameLift Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in de SDK do servidor do Unreal, consulte [Integre a Amazon GameLift em um projeto Unreal Engine](#).

Ações

- [GetSdkVersion\(\)](#)
- [InitSDK\(\)](#)
- [InitSDK\(\)](#)
- [ProcessReady\(\)](#)
- [ProcessEnding\(\)](#)
- [ActivateGameSession\(\)](#)
- [UpdatePlayerSessionCreationPolicy\(\)](#)
- [GetGameSessionId\(\)](#)
- [GetTerminationTime\(\)](#)
- [AcceptPlayerSession\(\)](#)
- [RemovePlayerSession\(\)](#)
- [DescribePlayerSessions\(\)](#)

- [StartMatchBackfill\(\)](#)
- [StopMatchBackfill\(\)](#)
- [GetComputeCertificate\(\)](#)
- [GetFleetRoleCredentials\(\)](#)

Note

Este tópico descreve a API Amazon GameLift C++ que você pode usar ao criar para o Unreal Engine. Especificamente, essa documentação se aplica ao código que você compila com a opção `-DBUILD_FOR_UNREAL=1`.

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
FGameLiftStringOutcome GetSdkVersion();
```

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto [the section called “FGameLiftStringOutcome”](#). O objeto retornada inclui o número da versão (exemplo, `5.0.0`). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =  
    Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma frota EC2 gerenciada. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método lê os parâmetros do servidor do ambiente host para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
FGameLiftGenericOutcome InitSDK()
```

Valor de retorno

Se bem-sucedido, retornará um objeto `InitSdkOutcome` indicando que o processo de servidor está pronto para chamar [ProcessReady\(\)](#).

Exemplo

```
//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK();
```

InitSDK()

Inicializa o Amazon GameLift SDK para uma Anywhere frota. Chame esse método na inicialização, antes que qualquer outra inicialização relacionada à Amazon GameLift ocorra. Esse método requer parâmetros explícitos do servidor para configurar a comunicação entre o servidor e o GameLift serviço da Amazon.

Sintaxe

```
FGameLiftGenericOutcome InitSDK(serverParameters)
```

Parâmetros

[F ServerParameters](#)

Para inicializar um servidor de jogo em uma GameLift Anywhere frota da Amazon, construa um `ServerParameters` objeto com as seguintes informações:

- O URL do WebSocket usado para se conectar ao seu servidor de jogo.
- O ID do processo usado para hospedar o servidor de jogos.
- O ID do computador que hospeda os processos do seu servidor de jogos.
- O ID da GameLift frota da Amazon que contém sua GameLift Anywhere computação da Amazon.

- O token de autorização gerado pela GameLift operação da Amazon.

Valor de retorno

Se bem-sucedido, retornará um objeto `InitSdkOutcome` indicando que o processo de servidor está pronto para chamar [ProcessReady\(\)](#).

Note

Se as chamadas para `InitSDK()` estiverem falhando para compilações de jogos implantadas em frotas do Anywhere, verifique o parâmetro `ServerSdkVersion` usado ao criar o recurso de compilação. Você deve definir explicitamente esse valor para a versão do SDK do servidor em uso. O valor padrão desse parâmetro é 4.x, o que não é compatível. Para resolver esse problema, crie uma nova versão e implante-a em uma nova frota.

Exemplo

```
//Define the server parameters
FServerParameters serverParameters;
parameters.m_authToken = "1111aaaa-22bb-33cc-44dd-5555eeee66ff";
parameters.m_fleetId = "arn:aws:gamelift:us-west-1:111122223333:fleet/
fleet-9999ffff-88ee-77dd-66cc-5555bbbb44aa";
parameters.m_hostId = "HardwareAnywhere";
parameters.m_processId = "PID1234";
parameters.m_webSocketUrl = "wss://us-west-1.api.amazongamelift.com";

//Call InitSDK to establish a local connection with the GameLift agent to enable
further communication.
FGameLiftGenericOutcome initSdkOutcome = gameLiftSdkModule->InitSDK(serverParameters);
```

ProcessReady()

Notifica a Amazon de GameLift que o processo do servidor está pronto para hospedar sessões de jogo. Chame esse método após invocar [InitSDK\(\)](#). Esse método deve ser chamado somente uma vez por processo.

Sintaxe

```
GenericOutcome ProcessReady(const Aws::GameLift::Server::ProcessParameters
&processParameters);
```

Parâmetros

processParameters

Um objeto [F ProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes dos métodos de retorno de chamada implementados no código do servidor do jogo que o GameLift serviço da Amazon invoca para se comunicar com o processo do servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho para qualquer arquivo específico da sessão de jogo que você deseja que a Amazon capture e GameLift armazene.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo ilustra as implementações das funções de chamada e delegação [ProcessReady\(\)](#).

```
//Calling ProcessReady tells GameLift this game server is ready to receive incoming
game sessions!
UE_LOG(GameServerLog, Log, TEXT("Calling Process Ready"));
FGameLiftGenericOutcome processReadyOutcome = gameLiftSdkModule-
>ProcessReady(*params);
```

ProcessEnding()

Notifica a Amazon de GameLift que o processo do servidor está sendo encerrado. Chame esse método depois de todas as outras tarefas de limpeza (incluindo o encerramento da sessão ativa do jogo) e antes de encerrar o processo. Dependendo do resultado `ProcessEnding()`, o processo sai com êxito (0) ou erro (-1) e gera um evento de frota. Se o processo for encerrado com um erro, o evento de frota gerado será `SERVER_PROCESS_TERMINATED_UNHEALTHY`.

Sintaxe

```
FGameLiftGenericOutcome ProcessEnding()
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

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

ActivateGameSession()

Notifica a Amazon de GameLift que o processo do servidor ativou uma sessão de jogo e agora está pronto para receber conexões de jogadores. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, após toda inicialização da sessão do jogo.

Sintaxe

```
FGameLiftGenericOutcome ActivateGameSession()
```

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo mostra o chamado `ActivateGameSession()` como parte da função de delegação `onStartGameSession()`.

```
//When a game session is created, GameLift sends an activation request to the game
server and passes along the game session object containing game properties and other
settings.
//Here is where a game server should take action based on the game session object.
//Once the game server is ready to receive incoming player connections, it should
invoke GameLiftServerAPI.ActivateGameSession()
auto onGameSession = [=](Aws::GameLift::Server::Model::GameSession gameSession)
{
```

```
FString gameId = FString(gameSession.GetGameSessionId());
UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameId);
gameLiftSdkModule->ActivateGameSession();
};
```

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador.

Sintaxe

```
FGameLiftGenericOutcome UpdatePlayerSessionCreationPolicy(EPlayerSessionCreationPolicy
policy)
```

Parâmetros

playerCreationSessionPolítica

Valor de string que indica se a sessão do jogo aceita novos jogadores.

Os valores válidos são:

- ACCEPT_ALL – Aceite todas as novas sessões de jogador.
- DENY_ALL – Recuse todas as novas sessões de jogador.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo define a política de ingresso da sessão do jogo atual para aceitar todos os jogadores.

```
FGameLiftGenericOutcome outcome = gameLiftSdkModule-
>UpdatePlayerSessionCreationPolicy(Aws::GameLift::Model::EPlayerSessionCreationPolicy::ACCEPT_A
```

GetGameSessionId()

Recupera o ID da sessão de jogo hospedada pelo processo do servidor ativo.

Para processos inativos que não são ativados com uma sessão de jogo, a chamada retorna um [the section called “F GameLiftError”](#).

Sintaxe

```
FGameLiftStringOutcome GetGameSessionId()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto [the section called “FGameLiftStringOutcome”](#). Se não for bem-sucedido, retornará uma mensagem de erro."

Para processos inativos que não são ativados com uma sessão de jogo, a chamada retorna `Success=True` e `GameSessionId=""`.

Exemplo

```
//When a game session is created, GameLift sends an activation request to the game
server and passes along the game session object containing game properties and other
settings.
//Here is where a game server should take action based on the game session object.
//Once the game server is ready to receive incoming player connections, it should
invoke GameLiftServerAPI.ActivateGameSession()
auto onGameSession = [](Aws::GameLift::Server::Model::GameSession gameSession)
{
    FString gameId = FString(gameSession.GetGameSessionId());
    UE_LOG(GameServerLog, Log, TEXT("GameSession Initializing: %s"), *gameId);
    gameLiftSdkModule->ActivateGameSession();
};
```

GetTerminationTime()

Retorna a hora em que um processo do servidor está programado para ser desligado, se essa informação estiver disponível. Um processo do servidor entra em ação após receber um `onProcessTerminate()` retorno de chamada da Amazon GameLift. A Amazon GameLift liga `onProcessTerminate()` pelos seguintes motivos:

- Quando o processo do servidor relatou problemas de saúde ou não respondeu à Amazon GameLift.
- Ao encerrar a instância durante um evento de redução.
- Quando uma instância é encerrada devido a uma [interrupção na instância spot](#).

Sintaxe

```
AwsDateTimeOutcome GetTerminationTime()
```

Valor de retorno

Se for bem-sucedido, retornará o horário de término como um objeto `AwsDateTimeOutcome`. O valor é o tempo de rescisão, expresso em tiques decorridos desde então `0001 00:00:00`. Por exemplo, o valor da data e hora `2020-09-13 12:26:40 -000Z` é igual aos tiques `637355968000000000`. Se nenhum horário de encerramento estiver disponível, o retornará uma mensagem de erro.

Se o processo não tiver recebido um retorno de chamada `ProcessParameters.OnProcessTerminate()`, uma mensagem de erro será retornada. Para obter mais informações sobre como desligar um processo do servidor, consulte [Responder a uma notificação de desligamento do processo do servidor](#).

Exemplo

```
AwsDateTimeOutcome TermTimeOutcome = gameLiftSdkModule->GetTerminationTime();
```

AcceptPlayerSession()

Notifica a Amazon de GameLift que um jogador com o ID de sessão de jogador especificado se conectou ao processo do servidor e precisa de validação. A Amazon GameLift verifica se o ID da sessão do jogador é válido. Depois que a sessão do jogador é validada, a Amazon GameLift altera o status do slot do jogador de RESERVADO para ATIVO.

Sintaxe

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerSessionId)
```

Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Este exemplo trata de uma solicitação de conexão que inclui a validação e a rejeição de IDs de sessão de jogadores inválidas.

```
bool GameLiftManager::AcceptPlayerSession(const FString& playerSessionId, const
FString& playerId)
{
    #if WITH_GAMELIFT
    UE_LOG(GameServerLog, Log, TEXT("Accepting GameLift PlayerSession: %s . PlayerId:
%s"), *playerSessionId, *playerId);
    FString gsId = GetCurrentGameSessionId();
    if (gsId.IsEmpty()) {
        UE_LOG(GameServerLog, Log, TEXT("No GameLift GameSessionId. Returning early!"));
        return false;
    }

    if (!gameLiftSdkModule->AcceptPlayerSession(playerSessionId).IsSuccess()) {
        UE_LOG(GameServerLog, Log, TEXT("PlayerSession not Accepted.));
        return false;
    }

    // Add PlayerSession from internal data structures keeping track of connected players
    connectedPlayerSessionIds.Add(playerSessionId);
    idToPlayerSessionMap.Add(playerSessionId, PlayerSession{ playerId,
playerSessionId });
    return true;
    #else
    return false;
    #endif
}
```

RemovePlayerSession()

Notifica a Amazon de GameLift que um jogador se desconectou do processo do servidor. Em resposta, a Amazon GameLift altera o slot do jogador para disponível.

Sintaxe

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerSessionId)
```


Parâmetros

playerSessionId

ID exclusivo emitido pela Amazon GameLift quando uma nova sessão de jogador é criada.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
bool GameLiftManager::RemovePlayerSession(const FString& playerSessionId)
{
    #if WITH_GAMELIFT
    UE_LOG(GameServerLog, Log, TEXT("Removing GameLift PlayerSession: %s"),
    *playerSessionId);

    if (!gameLiftSdkModule->RemovePlayerSession(playerSessionId).IsSuccess()) {
        UE_LOG(GameServerLog, Log, TEXT("PlayerSession Removal Failed"));
        return false;
    }

    // Remove PlayerSession from internal data structures that are keeping track of
    connected players
    connectedPlayerSessionIds.Remove(playerSessionId);
    idToPlayerSessionMap.Remove(playerSessionId);

    // end the session if there are no more players connected
    if (connectedPlayerSessionIds.Num() == 0) {
        EndSession();
    }

    return true;
    #else
    return false;
    #endif
}
```

DescribePlayerSessions()

Recupera dados da sessão do jogador, que incluem configurações, metadados da sessão e dados do jogador. Use esse método para obter informações sobre o seguinte:

- Uma sessão para um jogador
- Todas as sessões de jogador em uma sessão de jogo
- Todas as sessões de jogadores associadas a um único ID de jogador

Sintaxe

```
FGameLiftDescribePlayerSessionsOutcome DescribePlayerSessions(const
    FGameLiftDescribePlayerSessionsRequest &describePlayerSessionsRequest)
```

Parâmetros

[F GameLiftDescribePlayerSessionsRequest](#)

Um objeto [the section called “F GameLiftDescribePlayerSessionsRequest”](#) que descreve quais sessões de jogador recuperar.

Valor de retorno

Se bem-sucedido, retorna um objeto [the section called “F GameLiftDescribePlayerSessionsOutcome”](#) que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação.

Exemplo

Este exemplo solicita todas as sessões de jogador conectadas ativamente a uma sessão de jogo especificada. Ao omitir NextToken e definir o valor limite como 10, a Amazon GameLift retorna os primeiros registros de sessão de 10 jogadores correspondentes à solicitação.

```
void GameLiftManager::DescribePlayerSessions()
{
    #if WITH_GAMELIFT
    FString localPlayerSessions;
    for (auto& psId : connectedPlayerSessionIds)
    {
        PlayerSession ps = idToPlayerSessionMap[psId];
        localPlayerSessions += FString::Printf(TEXT("%s : %s ; "), *(ps.playerSessionId),
*(ps.playerId));
    }
    UE_LOG(GameServerLog, Log, TEXT("LocalPlayerSessions: %s"), *localPlayerSessions);
    #endif
}
```

```
UE_LOG(GameServerLog, Log, TEXT("Describing PlayerSessions in this GameSession"));
FGameLiftDescribePlayerSessionsRequest request;
request.m_gameSessionId = GetCurrentGameSessionId();

FGameLiftDescribePlayerSessionsOutcome outcome = gameLiftSdkModule-
>DescribePlayerSessions(request);
LogDescribePlayerSessionsOutcome(outcome);
#endif
}
```

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Esta ação é assíncrona. Se novos jogadores forem combinados, a Amazon GameLift fornece dados atualizados do matchmaker usando a função de retorno de chamada. `OnUpdateGameSession()`

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
FGameLiftStringOutcome StartMatchBackfill (FStartMatchBackfillRequest
&startBackfillRequest);
```

Parâmetros

[F StartMatchBackfillRequest](#)

Um `StartMatchBackfillRequest` objeto que comunica as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essas informações são opcionais; se nenhum ID for fornecido, a Amazon GameLift gerará um.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor está nos dados do marcador de jogos da sessão de jogo.
- O ID da sessão de jogo a ser preenchida.
- Os dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Valor de retorno

Retorna um objeto `StartMatchBackfillOutcome` com o ID do tíquete de alocação de correspondência ou um falha com uma mensagem de erro.

Exemplo

```

FGameLiftStringOutcome FGameLiftServerSDKModule::StartMatchBackfill(const
  FStartMatchBackfillRequest& request)
{
  #if WITH_GAMELIFT
  Aws::GameLift::Server::Model::StartMatchBackfillRequest sdkRequest;
  sdkRequest.SetTicketId(TCHAR_TO_UTF8(*request.m_ticketId));
  sdkRequest.SetGameSessionArn(TCHAR_TO_UTF8(*request.m_gameSessionArn));

  sdkRequest.SetMatchmakingConfigurationArn(TCHAR_TO_UTF8(*request.m_matchmakingConfigurationArn));
  for (auto player : request.m_players) {
    Aws::GameLift::Server::Model::Player sdkPlayer;
    sdkPlayer.SetPlayerId(TCHAR_TO_UTF8(*player.m_playerId));
    sdkPlayer.SetTeam(TCHAR_TO_UTF8(*player.m_team));
    for (auto entry : player.m_latencyInMs) {
      sdkPlayer.WithLatencyMs(TCHAR_TO_UTF8(*entry.Key), entry.Value);
    }

    std::map<std::string, Aws::GameLift::Server::Model::AttributeValue>
    sdkAttributeMap;
    for (auto attributeEntry : player.m_playerAttributes) {
      FAttributeValue value = attributeEntry.Value;
      Aws::GameLift::Server::Model::AttributeValue attribute;
      switch (value.m_type) {
        case FAttributeType::STRING:
          attribute =
            Aws::GameLift::Server::Model::AttributeValue(TCHAR_TO_UTF8(*value.m_S));
          break;
        case FAttributeType::DOUBLE:
          attribute = Aws::GameLift::Server::Model::AttributeValue(value.m_N);
          break;
        case FAttributeType::STRING_LIST:
          attribute =
            Aws::GameLift::Server::Model::AttributeValue::ConstructStringList();
          for (auto sl : value.m_SL) {
            attribute.AddString(TCHAR_TO_UTF8(*sl));
          };
          break;

```

```

        case FAttributeType::STRING_DOUBLE_MAP:
            attribute =
    Aws::GameLift::Server::Model::AttributeValue::ConstructStringDoubleMap();
            for (auto sdm : value.m_SDM) {
                attribute.AddStringAndDouble(TCHAR_TO_UTF8(*sdm.Key), sdm.Value);
            };
            break;
        }
        sdkPlayer.WithPlayerAttribute((TCHAR_TO_UTF8(*attributeEntry.Key)), attribute);
    }
    sdkRequest.AddPlayer(sdkPlayer);
}
auto outcome = Aws::GameLift::Server::StartMatchBackfill(sdkRequest);
if (outcome.IsSuccess()) {
    return FGameLiftStringOutcome(outcome.GetResult().GetTicketId());
}
else {
    return FGameLiftStringOutcome(FGameLiftError(outcome.GetError()));
}
#else
return FGameLiftStringOutcome("");
#endif
}
}

```

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa. Para obter mais informações, consulte o [recurso FlexMatch de preenchimento](#).

Sintaxe

```

FGameLiftGenericOutcome StopMatchBackfill (FStopMatchBackfillRequest
&stopBackfillRequest);

```

Parâmetros

[F StopMatchBackfillRequest](#)

Um StopMatchBackfillRequest objeto identificando o tíquete de matchmaking a ser cancelado:

- O ID do ticket atribuído à solicitação de alocação.
- O marcador de jogo que recebeu a solicitação de alocação.
- A sessão do jogo associada à solicitação de alocação.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

```
FGameLiftGenericOutcome FGameLiftServerSDKModule::StopMatchBackfill(const
FStopMatchBackfillRequest& request)
{
    #if WITH_GAMELIFT
    Aws::GameLift::Server::Model::StopMatchBackfillRequest sdkRequest;
    sdkRequest.SetTicketId(TCHAR_TO_UTF8(*request.m_ticketId));
    sdkRequest.SetGameSessionArn(TCHAR_TO_UTF8(*request.m_gameSessionArn));

    sdkRequest.SetMatchmakingConfigurationArn(TCHAR_TO_UTF8(*request.m_matchmakingConfigurationArn));
    auto outcome = Aws::GameLift::Server::StopMatchBackfill(sdkRequest);
    if (outcome.IsSuccess()) {
        return FGameLiftGenericOutcome(nullptr);
    }
    else {
        return FGameLiftGenericOutcome(FGameLiftError(outcome.GetError()));
    }
    #else
    return FGameLiftGenericOutcome(nullptr);
    #endif
}
```

GetComputeCertificate()

Recupera o caminho para o certificado TLS usado para criptografar a conexão de rede entre seu GameLift Anywhere recurso computacional da Amazon e a Amazon. GameLift Você pode usar o caminho do certificado ao registrar seu dispositivo computacional em uma GameLift Anywhere frota da Amazon. Para obter mais informações, consulte, [RegisterCompute](#).

Sintaxe

```
FGameLiftGetComputeCertificateOutcome FGameLiftServerSDKModule::GetComputeCertificate()
```

Valor de retorno

Retorna um objeto `GetComputeCertificateResponse` contendo o seguinte:

- `CertificatePath`: o caminho para o certificado TLS em seu recurso computacional.

- **HostName:** o nome do host do seu recurso computacional.

Exemplo

```
FGameLiftGetComputeCertificateOutcome FGameLiftServerSDKModule::GetComputeCertificate()
{
    #if WITH_GAMELIFT
    auto outcome = Aws::GameLift::Server::GetComputeCertificate();
    if (outcome.IsSuccess()) {
        auto& outres = outcome.GetResult();
        FGameLiftGetComputeCertificateResult result;
        result.m_certificate_path = UTF8_TO_TCHAR(outres.GetCertificatePath());
        result.m_computeName = UTF8_TO_TCHAR(outres.GetComputeName());
        return FGameLiftGetComputeCertificateOutcome(result);
    }
    else {
        return FGameLiftGetComputeCertificateOutcome(FGameLiftError(outcome.GetError()));
    }
    #else
    return FGameLiftGetComputeCertificateOutcome(FGameLiftGetComputeCertificateResult());
    #endif
}
```

GetFleetRoleCredentials()

Recupera as credenciais da função do IAM que autorizam a GameLift Amazon a interagir com outras. Serviços da AWS Para ter mais informações, consulte [Comunique-se com outros recursos AWS de suas frotas](#).

Sintaxe

```
FGameLiftGetFleetRoleCredentialsOutcome
FGameLiftServerSDKModule::GetFleetRoleCredentials(const
FGameLiftGetFleetRoleCredentialsRequest &request)
```

Parâmetros

[F GameLiftGetFleetRoleCredentialsRequest](#)

Valor de retorno

Informa um objeto [the section called “F GameLiftGetFleetRoleCredentialsOutcome”](#).

Exemplo

```
FGameLiftGetFleetRoleCredentialsOutcome
FGameLiftServerSDKModule::GetFleetRoleCredentials(const
FGameLiftGetFleetRoleCredentialsRequest &request)
{
    #if WITH_GAMELIFT
    Aws::GameLift::Server::Model::GetFleetRoleCredentialsRequest sdkRequest;
    sdkRequest.SetRoleArn(TCHAR_TO_UTF8(*request.m_roleArn));
    sdkRequest.SetRoleSessionName(TCHAR_TO_UTF8(*request.m_roleSessionName));

    auto outcome = Aws::GameLift::Server::GetFleetRoleCredentials(sdkRequest);


    if (outcome.IsSuccess()) {
        auto& outres = outcome.GetResult();
        FGameLiftGetFleetRoleCredentialsResult result;
        result.m_assumedUserRoleArn = UTF8_TO_TCHAR(outres.GetAssumedUserRoleArn());
        result.m_assumedRoleId = UTF8_TO_TCHAR(outres.GetAssumedRoleId());
        result.m_accessKeyId = UTF8_TO_TCHAR(outres.GetAccessKeyId());
        result.m_secretAccessKey = UTF8_TO_TCHAR(outres.GetSecretAccessKey());
        result.m_sessionToken = UTF8_TO_TCHAR(outres.GetSessionToken());
        result.m_expiration = FDateTime::FromUnixTimestamp(outres.GetExpiration());
        return FGameLiftGetFleetRoleCredentialsOutcome(result);
    }
    else {
        return FGameLiftGetFleetRoleCredentialsOutcome(FGameLiftError(outcome.GetError()));
    }
    #else
    return
    FGameLiftGetFleetRoleCredentialsOutcome(FGameLiftGetFleetRoleCredentialsResult());
    #endif
}
```

Referência GameLift do Amazon Server SDK (Unreal): tipos de dados

Você pode usar essa referência do SDK do servidor Amazon GameLift Unreal para ajudá-lo a preparar seu jogo multijogador para uso com a Amazon. GameLift Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#) e para obter informações sobre como usar o plug-in do SDK do servidor do Unreal, consulte [Integre a Amazon GameLift em um projeto Unreal Engine](#).

Tipos de dados

- [F ProcessParameters](#)
- [UpdateGameSession](#)
- [GameSession](#)
- [F ServerParameters](#)
- [F StartMatchBackfillRequest](#)
- [FPlayer](#)
- [F GameLiftDescribePlayerSessionsRequest](#)
- [F StopMatchBackfillRequest](#)
- [F AttributeValue](#)
- [F GameLiftGetFleetRoleCredentialsRequest](#)
- [F GameLiftLongOutcome](#)
- [F GameLiftStringOutcome](#)
- [F GameLiftDescribePlayerSessionsOutcome](#)
- [F GameLiftDescribePlayerSessionsResult](#)
- [F GenericOutcome](#)
- [F GameLiftPlayerSession](#)
- [F GameLiftGetComputeCertificateOutcome](#)
- [F GameLiftGetComputeCertificateResult](#)
- [F GameLiftGetFleetRoleCredentialsOutcome](#)
- [F GetFleetRoleCredentialsResult](#)
- [F GameLiftError](#)
- [Enumerações](#)

 Note

Este tópico descreve a API Amazon GameLift C++ que você pode usar ao criar para o Unreal Engine. Especificamente, essa documentação se aplica ao código que você compila com a opção `-DBUILD_FOR_UNREAL=1`.

F ProcessParameters

Esse tipo de dados contém o conjunto de parâmetros enviados para a Amazon GameLift em um [ProcessReady\(\)](#).

Properties	Descrição
LogParameters	<p>Um objeto com caminhos de diretório para arquivos que são gerados durante uma sessão de jogo. A Amazon GameLift copia e armazena os arquivos para acesso futuro.</p> <p>Digite: TArray<FString></p> <p>Obrigatório: não</p>
OnHealthCheck	<p>A função de retorno de chamada que a Amazon GameLift invoca para solicitar um relatório de status de saúde do processo do servidor. A Amazon GameLift chama essa função a cada 60 segundos e espera 60 segundos por uma resposta. O processo do servidor retorna TRUE se estiver íntegro e FALSE se não estiver íntegro. Se nenhuma resposta for retornada, a Amazon GameLift registra o processo do servidor como não íntegro.</p> <p>Essa propriedade é uma função delegada definida como DECLARE_DELEGATE_R etVal(bool, FOnHealthCheck) ;</p> <p>Digite: FOnHealthCheck</p> <p>Obrigatório: não</p>
OnProcessTerminate	<p>A função de retorno de chamada que a Amazon GameLift invoca para forçar o encerramento do processo do servidor. Depois</p>

de chamar essa função, a Amazon GameLift espera 5 minutos até que o processo do servidor seja encerrado e responde com uma [ProcessEnding\(\)](#) chamada antes de encerrar o processo do servidor.

Digite: FSimpleDelegate

Obrigatório: Sim

OnStartGameSession

A função de retorno de chamada que a Amazon GameLift invoca para ativar uma nova sessão de jogo. A Amazon GameLift chama essa função em resposta a uma solicitação do cliente [CreateGameSession](#). A função de retorno de chamada passa um [GameSession](#) objeto, conforme definido na Amazon GameLift API Reference.

Essa propriedade é uma função delegada definida como `DECLARE_DELEGATE_OneParam(FOnStartGameSession, Aws::GameLift::Server::Mode1::GameSession);`

Digite: FOnStartGameSession

Obrigatório: Sim

OnUpdateGameSession

A função de retorno de chamada que a Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. A Amazon GameLift chama essa função quando uma solicitação de preenchimento de correspondência é processada para fornecer dados atualizados do matchmaker. Ele passa um [GameSession](#) objeto, uma atualização de status (updateReason) e o ID do tíquete de preenchimento correspondente.

Essa propriedade é uma função delegada definida como DECLARE_DELEGATE_OnUpdateGameSessionParam(FOnUpdateGameSession, Aws::GameLift::Server::Model::UpdateGameSession);

Digite: FOnUpdateGameSession

Obrigatório: não

Porta

O número da porta em que o processo de servidor escuta para novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Digite: int

Obrigatório: Sim

UpdateGameSession

Esse tipo de dados é atualizado para um objeto de sessão de jogo, o que inclui o motivo pelo qual a sessão de jogo foi atualizada e o ID do tíquete de alocação relacionada, se a alocação for usada para preencher as sessões dos jogadores na sessão do jogo.

Propriedades	Descrição
GameSession	<p>Um GameSession objeto definido pela GameLift API da Amazon. O objeto <code>GameSession</code> contém propriedades que descrevem uma sessão de jogo.</p> <p>Digite: <code>Aws::GameLift::Server::GameSession</code></p> <p>Obrigatório: não</p>
UpdateReason	<p>O motivo pelo qual a sessão do jogo está sendo atualizada.</p> <p>Digite: <code>enum class UpdateReason</code></p> <ul style="list-style-type: none">• <code>MATCHMAKING_DATA_UPDATED</code>• <code>BACKFILL_FAILED</code>• <code>BACKFILL_TIMED_OUT</code>• <code>BACKFILL_CANCELLED</code> <p>Obrigatório: não</p>
BackfillTicketId	<p>O ID do tíquete de alocação que está tentando atualizar a sessão do jogo.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>

GameSession

Esse tipo de dados fornece detalhes de uma sessão de jogo.

Propriedades	Descrição
GameSessionId	<p>Um identificador exclusivo da sessão de jogo. Um ARN de sessão de jogo tem o seguinte formato: <code>arn:aws:gamelift:<region>::gamesession/<fleet ID>/<custom ID string or idempotency token></code> .</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>
Nome	<p>Um rótulo descritivo da sessão do jogo.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>
FleetId	<p>Um identificador exclusivo da frota em que a sessão de jogo está sendo executada.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>
MaximumPlayerSessionCount	<p>O número máximo de conexões do jogador com a sessão do jogo.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>
Porta	<p>O número da porta para a sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta.</p>

Propriedades	Descrição
	Digite: <code>int</code> Obrigatório: não
<code>IpAddress</code>	O endereço IP da sessão de jogo. Para se conectar a um servidor de GameLift jogos da Amazon, um aplicativo precisa do endereço IP e do número da porta. Digite: <code>char[]</code> Obrigatório: não
<code>GameSessionData</code>	Um conjunto de propriedades de sessão de jogo personalizadas, formatadas como um único valor de string. Digite: <code>char[]</code> Obrigatório: não
<code>MatchmakerData</code>	Informações sobre o processo de marcação de jogos usado para criar a sessão do jogo, na sintaxe JSON, formatado como uma string. Além da configuração de marcação de jogos usada, ele contém dados sobre todos os jogadores atribuídos à partida, incluindo atributos do jogador e atribuições da equipe. Digite: <code>char[]</code> Obrigatório: não

Propriedades	Descrição
GameProperties	<p>Um conjunto de propriedades personalizadas para uma sessão de jogo, formatado como pares de chave:valor. Essas propriedades são passadas com uma solicitação de início de uma nova sessão de jogo.</p> <p>Digite: <code>GameProperty[]</code></p> <p>Obrigatório: não</p>
DnsName	<p>O identificador DNS atribuído à instância que está executando a sessão do jogo. Os valores têm o formato a seguir:</p> <ul style="list-style-type: none"> Frotas habilitadas para TLS: <code><unique identifier>.<region identifier>.amazongamelift.com</code> Frotas não habilitadas para TLS: <code>ec2-unique identifier>.compute.amazonaws.com</code> <p>Ao se conectar a uma sessão de jogo que está sendo executada em uma frota compatível com TLS, você deve usar o nome DNS, não o endereço IP.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: não</p>

F ServerParameters

Informações usadas para manter a conexão entre um GameLift Anywhere servidor da Amazon e o GameLift serviço da Amazon. Essas informações são usadas ao iniciar novos processos de servidor com [InitSDK\(\)](#). Para servidores hospedados em instâncias EC2 GameLift gerenciadas pela Amazon, use um objeto vazio.

Propriedades	Descrição
websocketUrl	<p>A <code>GameLiftServerSdkEndpoint</code> Amazon GameLift retorna quando você compra RegisterCompute um recurso GameLift Anywhere computacional da Amazon.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: Sim</p>
processId	<p>Um identificador exclusivo registrado no processo do servidor que hospeda seu jogo.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: Sim</p>
hostId	<p>O <code>HostID</code> é o <code>ComputeName</code> usado quando você registrou seu computador. Para obter mais informações, consulte, RegisterCompute.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: Sim</p>
fleetId	<p>O identificador exclusivo da frota na qual o computador está registrado. Para obter mais informações, consulte, RegisterCompute.</p> <p>Digite: <code>char[]</code></p> <p>Obrigatório: Sim</p>
authToken	<p>O token de autenticação gerado pela Amazon GameLift que autentica seu servidor na Amazon GameLift. Para obter mais informações, consulte, GetComputeAuthToken.</p> <p>Digite: <code>char[]</code></p>

Propriedades	Descrição
	Obrigatório: Sim

F StartMatchBackfillRequest

Informações usadas para criar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações à Amazon GameLift em uma [StartMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	Um identificador da sessão de jogo exclusivo . A operação de API retorna GetGameSessionId o identificador no formato de ARN. Digite: <code>char[]</code> Obrigatório: Sim
MatchmakingConfigurationArn	Um identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. O marcador de jogos ARN da sessão de jogo original está no objeto de sessão de jogo na propriedade de dados do marcador de jogos. Saiba mais sobre os dados do marcador de jogos em Trabalhar com dados do marcador de jogos . Digite: <code>char[]</code> Obrigatório: Sim
Jogadores	Um conjunto de dados que representa todos os jogadores que estão na sessão de jogo. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais.

Propriedades	Descrição
	Digite: TArray<FPlayer> Obrigatório: Sim
TicketId	Um identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se você não fornecer um valor, a Amazon GameLift gera um. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário. Digite: char[] Obrigatório: não

FPlayer

Esse tipo de dados representa um jogador em marcação de jogos. Ao iniciar uma solicitação de marcação de jogos, um jogador tem um ID de jogador, atributos, e possivelmente dados de latência. A Amazon GameLift adiciona informações da equipe após a realização de uma partida.

Propriedades	Descrição
LatencyInMS	Um conjunto de valores expressos em milissegundos que indicam a quantidade de latência que um jogador experimenta quando conectado a um local. Se essa propriedade for usada, o jogador só será correspondido aos locais listados. Se um marcador de jogos tiver uma regra que avalie a latência do jogador, os jogadores deverão reportar a latência a ser correspondida. Digite: TMap>FString, int32<

Propriedades	Descrição
	Obrigatório: não
PlayerAttributes	<p>Uma coleção de pares de chave:valor contendo informações de jogadores para uso em jogos de jogos. As chaves de atributos do jogador devem corresponder às PlayerAttributes usadas em um conjunto de regras de matchmaking.</p> <p>Para obter mais informações sobre os atributos do jogador, consulte AttributeValue.</p> <p>Digite: TMap>FString, FAttributeValue<</p> <p>Obrigatório: não</p>
PlayerId	<p>Um identificador exclusivo para um jogador.</p> <p>Digite: std::string</p> <p>Obrigatório: não</p>
Team	<p>O nome da equipe à qual o jogador está designado em uma partida. Você define o nome da equipe no conjunto de regras de marcação de jogos.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>

F GameLiftDescribePlayerSessionsRequest

Um objeto que especifica quais sessões de jogador devem ser recuperadas. O processo do servidor fornece essas informações com uma [DescribePlayerSessions\(\)](#) chamada para a Amazon GameLift.

Propriedades	Descrição
GameSessionId	<p>Um identificador da sessão de jogo exclusivo . Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada.</p> <p>O formato do ID da sessão do jogo é FString. A GameSessionID é uma string de ID personalizada ou uma</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
PlayerSessionId	<p>Um identificador exclusivo para uma sessão de jogador. Use esse parâmetro para solicitar uma única sessão de jogador específica.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>
PlayerId	<p>Um identificador exclusivo para um jogador. Use este parâmetro para solicitar todas as sessões de jogador para um jogador específico. Consulte Gerar IDs de jogador.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>
PlayerSessionStatusFilter	<p>O status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis incluem:</p> <ul style="list-style-type: none">RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor nem foi validado.

Propriedades	Descrição
	<ul style="list-style-type: none"> • ACTIVE – O player foi validado pelo processo do servidor e está conectado. • COMPLETED – A conexão do jogador foi interrompida. • TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou ou não foi validado dentro do tempo limite (60 segundos). <p>Digite: <code>FString</code></p> <p>Obrigatório: não</p>
NextToken	<p>O token que indica o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>FString</code></p> <p>Obrigatório: não</p>
Limite	<p>O número máximo de resultados a serem retornados. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado.</p> <p>Digite: <code>int</code></p> <p>Obrigatório: não</p>

F StopMatchBackfillRequest

Informações usadas para cancelar uma solicitação de alocação de marcação de jogos. O servidor do jogo comunica essas informações ao GameLift serviço da Amazon em uma [StopMatchBackfill\(\)](#) chamada.

Propriedades	Descrição
GameSessionArn	Um identificador exclusivo da sessão de jogo da solicitação que está sendo cancelada. Digite: FString Obrigatório: Sim
MatchmakingConfigurationArn	Um identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada. Digite: FString Obrigatório: Sim
TicketId	Um identificador exclusivo do tíquete de solicitação de alocação a ser cancelado. Digite: FString Obrigatório: Sim

F AttributeValue

Use esses valores em pares de chave-valor de atributo [FPlayer](#). Esse objeto permite especificar um valor de atributo usando qualquer um dos tipos de dados válidos: string, número, matriz de string ou mapa de dados. Cada objeto `AttributeValue` pode usar somente uma das propriedades disponíveis.

Propriedades	Descrição
attrType	Especifica o tipo de valor do atributo. Tipo: Um <code>FAttributeType</code> valor enum.. Obrigatório: não
S	Representa um valor de atributo de string.

Propriedades	Descrição
	Digite: <code>FString</code> Obrigatório: não
N	Representa um valor de atributo numérico. Digite: <code>double</code> Obrigatório: não
SL	Representa uma matriz de valores de atributos de string. Digite: <code>TArray<FString></code> Obrigatório: não
SDM	Representa um dicionário de chaves de string e valores duplos. Digite: <code>TMap<FString, double></code> Obrigatório: não

F GameLiftGetFleetRoleCredentialsRequest

Esse tipo de dados fornece credenciais de função que ampliam o acesso limitado aos seus recursos AWS no servidor do jogo. Para obter mais informações, consulte, [Configurar uma função de serviço do IAM para a Amazon GameLift](#).

Propriedades	Descrição
RoleArn	O nome do recurso da Amazon (ARN) da perfil de serviço que estende o acesso limitado aos seus recursos AWS. Digite: <code>FString</code>

Propriedades	Descrição
	Obrigatório: não
RoleSessionName	O nome da sessão que descreve o uso das credenciais da função. Digite: FString Obrigatório: não

F GameLiftLongOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: long Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto. Digite: long&& Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “F GameLiftError”

Propriedades	Descrição
	Obrigatório: não

F GameLiftStringOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: FString Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto. Digite: FString&& Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “F GameLiftError” Obrigatório: não

F GameLiftDescribePlayerSessionsOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	<p>O resultado da ação.</p> <p>Digite: the section called “F GameLiftDescribePlayerSessionsResult”</p> <p>Obrigatório: não</p>
ResultWithOwnership	<p>O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto.</p> <p>Digite: <code>FGameLiftDescribePlayerSessionsResult&&</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “F GameLiftError”</p> <p>Obrigatório: não</p>

F GameLiftDescribePlayerSessionsResult

Propriedades	Descrição
PlayerSessions	<p>Digite: <code>TArray<FGameLiftPlayerSession></code></p>

Propriedades	Descrição
	Obrigatório: Sim
NextToken	O token indicando o início da próxima página de resultados. Para especificar o início do conjunto de resultados, não forneça um valor. Se fornecer um ID de sessão do jogador, esse parâmetro será ignorado. Digite: FString Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “F GameLiftError” Obrigatório: não

F GenericOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: bool Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida.

Propriedades	Descrição
	Digite: the section called “F GameLiftError”
	Obrigatório: não

F GameLiftPlayerSession

Propriedades	Descrição
CreationTime	Digite: long Obrigatório: Sim
FleetId	Digite: FString Obrigatório: Sim
GameSessionId	Digite: FString Obrigatório: Sim
IpAddress	Digite: FString Obrigatório: Sim
PlayerData	Digite: FString Obrigatório: Sim
PlayerId	Digite: FString Obrigatório: Sim
PlayerSessionId	Digite: FString Obrigatório: Sim
Porta	Digite: int Obrigatório: Sim

Propriedades	Descrição
Status	Tipo: Um <code>PlayerSessionStatus</code> enum.. Obrigatório: Sim
TerminationTime	Digite: <code>long</code> Obrigatório: Sim
DnsName	Digite: <code>FString</code> Obrigatório: Sim

F GameLiftGetComputeCertificateOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “F GameLiftGetComputeCertificateResult” Obrigatório: não
ResultWithOwnership	O resultado da ação, convertido como um <code>rvalue</code> , para que o código de chamada possa se apropriar do objeto. Digite: <code>FGameLiftGetComputeCertificateResult&&</code> Obrigatório: não
Bem-sucedida	Se a ação foi bem-sucedida ou não. Digite: <code>bool</code>

Propriedades	Descrição
	Obrigatório: Sim
Erro	O erro que ocorreu se a ação não foi bem-sucedida. Digite: the section called “F GameLiftError” Obrigatório: não

F GameLiftGetComputeCertificateResult

O caminho para o certificado TLS em seu computador e o nome do host do computador.

Propriedades	Descrição
CertificatePath	Digite: FString Obrigatório: Sim
ComputeName	Digite: FString Obrigatório: Sim

F GameLiftGetFleetRoleCredentialsOutcome

Esse tipo de dados resulta de uma ação e produz um objeto com as seguintes propriedades:

Propriedades	Descrição
Resultado	O resultado da ação. Digite: the section called “F GetFleetRoleCredentialsResult” Obrigatório: não

Propriedades	Descrição
ResultWithOwnership	<p>O resultado da ação, convertido como um rvalue, para que o código de chamada possa se apropriar do objeto.</p> <p>Digite: <code>FGameLiftGetFleetRoleCredentialsResult</code></p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: <code>bool</code></p> <p>Obrigatório: Sim</p>
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “F GameLiftError”</p> <p>Obrigatório: não</p>

F GetFleetRoleCredentialsResult

Propriedades	Descrição
AccessKeyId	<p>O ID da chave de acesso para autenticar e fornecer acesso aos seus recursos AWS.</p> <p>Digite: <code>FString</code></p> <p>Obrigatório: não</p>
AssumedRoleId	<p>O ID do usuário ao qual o perfil de serviço pertence.</p> <p>Digite: <code>FString</code></p>

Propriedades	Descrição
	Obrigatório: não
AssumedRoleUserArn	<p>O nome do recurso da Amazon (ARN) do usuário ao perfil de serviço de serviço pertence.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>
Expiração	<p>A quantidade de tempo até que suas credenciais de sessão expirem.</p> <p>Digite: FDateTime</p> <p>Obrigatório: não</p>
SecretAccessKey	<p>O ID da chave de acesso secreta para autenticação.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>
SessionToken	<p>Um token para identificar a sessão ativa atual interagindo com seus recursos AWS.</p> <p>Digite: FString</p> <p>Obrigatório: não</p>
Bem-sucedida	<p>Se a ação foi bem-sucedida ou não.</p> <p>Digite: bool</p> <p>Obrigatório: Sim</p>

Propriedades	Descrição
Erro	<p>O erro que ocorreu se a ação não foi bem-sucedida.</p> <p>Digite: the section called “GameLiftError”</p> <p>Obrigatório: não</p>

F GameLiftError

Propriedades	Descrição
ErrorType	<p>O tipo de erro.</p> <p>Tipo: Um GameLiftErrorType enum.</p> <p>Obrigatório: não</p>
ErrorMessage	<p>A mensagem de erro.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>
ErrorName	<p>O nome do erro.</p> <p>Digite: <code>std::string</code></p> <p>Obrigatório: não</p>

Enumerações

Os enums definidos para o SDK GameLift do servidor Amazon (Unreal) são definidos da seguinte forma:

F AttributeType

- NONE
- STRING

- DOUBLE
- STRING_LIST
- STRING_DOUBLE_MAP

GameLiftErrorType

Valor da string indicando o tipo de erro. Os valores válidos são:

- SERVICE_CALL_FAILED – Uma chamada AWS para um serviço falhou.
- LOCAL_CONNECTION_FAILED — A conexão local com a Amazon falhou. GameLift
- NETWORK_NOT_INITIALIZED – A rede não foi inicializada.
- GAMESESSION_ID_NOT_SET – O ID da sessão do jogo não foi definido.
- BAD_REQUEST_EXCEPTION
- INTERNAL_SERVICE_EXCEPTION
- ALREADY_INITIALIZED — O servidor ou cliente GameLift da Amazon já foi inicializado com Initialize ().
- FLEET_MISMATCH – A frota alvo não corresponde à frota de uma GameSession ou PlayerSession.
- GAMELIFT_CLIENT_NOT_INITIALIZED — O cliente Amazon não foi inicializado. GameLift
- GAMELIFT_SERVER_NOT_INITIALIZED — O servidor Amazon não foi inicializado. GameLift
- GAME_SESSION_ENDED_FAILED — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que a sessão do jogo foi encerrada.
- GAME_SESSION_NOT_READY — A sessão de jogo do GameLift Amazon Server não foi ativada.
- GAME_SESSION_READY_FAILED — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que a sessão do jogo está pronta.
- INITIALIZATION_MISMATCH – Um método de cliente foi chamado depois de Server::Initialize() ou vice-versa.
- NOT_INITIALIZED — O GameLift servidor ou cliente da Amazon não foi inicializado com Initialize ().
- NO_TARGET_ALIASID_SET – Um aliasID de destino não foi definido.
- NO_TARGET_FLEET_SET – Uma frota de destino não foi definida.
- PROCESS_ENDING_FAILED — O GameLift Amazon Server SDK não conseguiu entrar em contato com o serviço para informar que o processo está sendo encerrado.

- `PROCESS_NOT_ACTIVE` — O processo do servidor ainda não está ativo, não está vinculado a um e não pode `GameSession` aceitar ou processar. `PlayerSessions`
- `PROCESS_NOT_READY` – O processo do servidor ainda não está pronto para ser ativado.
- `PROCESS_READY_FAILED` — O `GameLift Amazon Server SDK` não conseguiu entrar em contato com o serviço para informar que o processo está pronto.
- `SDK_VERSION_DETECTION_FAILED` – Falha na detecção da versão do SDK.
- `STX_CALL_FAILED` – Uma chamada para o componente de back-end do servidor `xSTx` falhou.
- `STX_INITIALIZATION_FAILED` – O componente de back-end do servidor `XSTx` falhou ao inicializar.
- `UNEXPECTED_PLAYER_SESSION` – Uma sessão de jogador não registrada foi encontrada pelo servidor.
- `WEBSOCKET_CONNECT_FAILURE`
- `WEBSOCKET_CONNECT_FAILURE_FORBIDDEN`
- `WEBSOCKET_CONNECT_FAILURE_INVALID_URL`
- `WEBSOCKET_CONNECT_FAILURE_TIMEOUT`
- `WEBSOCKET_RETRIABLE_SEND_MESSAGE_FAILURE` — Falha recuperável ao enviar uma mensagem ao Serviço. `GameLift WebSocket`
- `WEBSOCKET_SEND_MESSAGE_FAILURE` — Falha ao enviar uma mensagem para o serviço. `GameLift WebSocket`
- `MATCH_BACKFILL_REQUEST_VALIDATION` – A validação da solicitação falhou.
- `PLAYER_SESSION_REQUEST_VALIDATION` – A validação da solicitação falhou.

E `PlayerSessionCreationPolicy`

Valor de string que indica se a sessão do jogo aceita novos jogadores. Os valores válidos são:

- `ACCEPT_ALL` – Aceite todas as novas sessões de jogador.
- `DENY_ALL` – Recuse todas as novas sessões de jogador.
- `NOT_SET` – A sessão do jogo não está configurada para aceitar ou negar sessões de novos jogadores.

E `PlayerSessionStatus`

- `ACTIVE`
- `COMPLETED`

- NOT_SET
- RESERVED
- TIMEDOUT

Referência do SDK 3.x do servidor do Amazon GameLift para Unreal Engine

Você poderá usar essa referência do SDK 3.x do servidor do Amazon GameLift para Unreal Engine para ajudá-lo a preparar seu jogo multijogador para uso com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Tópicos

- [Referência do SDK do servidor do Amazon GameLift para Unreal Engine: Actions](#)
- [Referência do SDK de servidor do Amazon GameLift para Unreal Engine: tipos de dados](#)

Referência do SDK do servidor do Amazon GameLift para Unreal Engine: Actions

Esta referência de Server SDK do Amazon GameLift pode ajudar a preparar os projetos de jogo Unreal Engine a serem usados com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Esta API é definida em `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Para configurar o plug-in do Unreal Engine e ver exemplos de código [Integre a Amazon GameLift em um projeto Unreal Engine](#).

- Ações
- [Tipos de dados](#)

AcceptPlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se conectou ao processo do servidor e precisa de validação. O Amazon GameLift verifica se o ID da sessão do jogador é válido, ou seja, se o ID do jogador reservou um slot de jogador na sessão do jogo. Depois de validado, o Amazon GameLift altera o status do slot do jogador de RESERVED para ACTIVE.

Sintaxe

```
FGameLiftGenericOutcome AcceptPlayerSession(const FString& playerId)
```

Parâmetros

playerSessionId

ID exclusiva emitida pelo serviço Amazon GameLift em resposta a uma chamada para a ação [CreatePlayerSession](#) da API do AWS SDK do Amazon GameLift. O cliente do jogo faz referência a esse ID ao se conectar ao processo do servidor.

Tipo: FString

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

ActivateGameSession()

Notifica o serviço do Amazon GameLift de que o processo do servidor ativou uma sessão do jogo e já está pronto para receber as conexões do jogador. Essa ação deve ser chamada como parte da função de retorno de chamada `onStartGameSession()`, depois que toda a inicialização da sessão tiver sido concluída.

Sintaxe

```
FGameLiftGenericOutcome ActivateGameSession()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

DescribePlayerSessions()

Recupera dados da sessão do jogador, inclusive configurações, metadados da sessão e dados do jogador. Use essa ação para obter informações de uma única sessão de jogador, para todas as

sessões de jogador em uma sessão de jogo, ou para todas as sessões de jogador associadas a um único ID de jogador.

Sintaxe

```
FGameLiftDescribePlayerSessionsOutcome DescribePlayerSessions(const  
    FGameLiftDescribePlayerSessionsRequest &describePlayerSessionsRequest)
```

Parâmetros

describePlayerSessionsRequest

Um objeto [FDescribePlayerSessionsRequest](#) que descreve quais sessões de jogador recuperar.

Obrigatório: Sim

Valor de retorno

Se bem-sucedido, retorna um objeto [FDescribePlayerSessionsRequest](#) que contém um conjunto de objetos de sessão do jogador que atendem aos parâmetros de solicitação. Os objetos de sessão de jogador têm uma estrutura idêntica à do tipo de dados [PlayerSession](#) da API do AWS SDK do Amazon GameLift.

GetGameSessionId()

Recupera o ID da sessão do jogo hospedada no momento pelo processo do servidor, caso o processo do servidor esteja ativo.

Sintaxe

```
FGameLiftStringOutcome GetGameSessionId()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará o ID da sessão de jogo como um objeto `FGameLiftStringOutcome`. Se não for bem-sucedido, retornará uma mensagem de erro.

GetInstanceCertificate()

Recupera a localização do arquivo de um certificado TLS codificado por pem que está associado à frota e suas instâncias. O AWS Certificate Manager gera esse certificado quando você cria uma nova frota com a configuração do certificado definida como GENERATED. Use esse certificado para estabelecer uma conexão segura com um cliente de jogo e para criptografar a comunicação entre cliente e servidor.

Sintaxe

```
FGameLiftGetInstanceCertificateOutcome GetInstanceCertificate()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se for bem-sucedido, o retorno será um objeto `GetInstanceCertificateOutcome` que contém a localização do arquivo de certificado TLS e da cadeia de certificados da frota, que estão armazenados na instância. Um arquivo de certificado raiz, extraído da cadeia de certificados, também é armazenado na instância. Se não for bem-sucedido, retornará uma mensagem de erro.

Para obter mais informações sobre o certificado e os dados da cadeia de certificados, consulte [Elementos de resposta GetCertificate](#) na Referência da API do AWS Certificate Manager.

GetSdkVersion()

Retorna o número da versão atual do SDK compilado no processo de servidor.

Sintaxe

```
FGameLiftStringOutcome GetSdkVersion();
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Se bem-sucedido, retornará a versão do SDK atual como um objeto `FGameLiftStringOutcome`. A string retornada inclui apenas o número da versão (exemplo, "3.1.5"). Se não for bem-sucedido, retornará uma mensagem de erro.

Exemplo

```
Aws::GameLift::AwsStringOutcome SdkVersionOutcome =  
    Aws::GameLift::Server::GetSdkVersion();
```

InitSDK()

Inicializa o SDK do Amazon GameLift. Esse método deve ser chamado na inicialização antes de qualquer outra inicialização relacionada ao Amazon GameLift.

Sintaxe

```
FGameLiftGenericOutcome InitSDK()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

ProcessEnding()

Notifica o serviço do Amazon GameLift de que o processo do servidor está sendo desligado. Esse método deverá ser chamado depois de todas as outras tarefas de limpeza, inclusive desligar todas as sessões de jogos ativas. Este método deve sair com um código de saída zero; um código de saída diferente de zero resulta em uma mensagem de evento indicando que o processo não foi encerrado corretamente.

Sintaxe

```
FGameLiftGenericOutcome ProcessEnding()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

ProcessReady()

Notifica o serviço do Amazon GameLift de que o processo do servidor está pronto para hospedar sessões do jogo. Chame esse método depois de invocar [InitSDK\(\)](#) e concluir todas as tarefas de configuração necessárias antes que o processo do servidor possa hospedar uma sessão do jogo. Esse método deve ser chamado somente uma vez por processo.

Sintaxe

```
FGameLiftGenericOutcome ProcessReady(FProcessParameters &processParameters)
```

Parâmetros

FProcessParameters

Um objeto [FProcessParameters](#) que fornece as seguintes informações sobre o processo do servidor:

- Nomes de métodos de retorno de chamada, implementados no código do servidor de jogos, que o serviço do Amazon GameLift invoca para se comunicar com o processo de servidor.
- Número da porta em que o processo de servidor está escutando.
- Caminho de qualquer arquivo específico da sessão do jogo que você deseja que o Amazon GameLift capture e armazene.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Exemplo

Consulte o código de exemplo em [Usar o plug-in do Unreal Engine](#).

RemovePlayerSession()

Notifica o serviço do Amazon GameLift de que um jogador com o ID da sessão do jogador especificado se desconectou do processo do servidor. Em resposta, o Amazon GameLift altera o slot do jogador para um disponível, o que permite ser atribuído a um novo jogador.

Sintaxe

```
FGameLiftGenericOutcome RemovePlayerSession(const FString& playerId)
```

Parâmetros

playerSessionId

[ID exclusiva emitida pelo serviço Amazon GameLift em resposta a uma chamada para a ação `CreatePlayerSession`](#) da API AWS SDK do Amazon GameLift. O cliente do jogo faz referência a esse ID ao se conectar ao processo do servidor.

Tipo: FString

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

StartMatchBackfill()

Envia uma solicitação para encontrar novos jogadores para os slots abertos em uma sessão de jogo criada com o FlexMatch. Consulte também a ação [StartMatchBackfill\(\)](#) do SDK da AWS. Com essa ação, as solicitações de alocação de correspondência podem ser iniciadas por um processo do servidor de jogos que esteja hospedando a sessão do jogo. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Esta ação é assíncrona. Se a correspondência dos novos jogadores for bem-sucedida, o serviço do Amazon GameLift entregará os dados atualizados do marcador de jogos usando a função de retorno de chamada `OnUpdateGameSession()`.

Um processo de servidor pode ter apenas uma solicitação de alocação de correspondência ativa por vez. Para enviar uma nova solicitação, primeiro chame [StopMatchBackfill\(\)](#) para cancelar a solicitação original.

Sintaxe

```
FGameLiftStringOutcome StartMatchBackfill (FStartMatchBackfillRequest  
&startBackfillRequest);
```

Parâmetros

FStartMatchBackfillRequest

Um objeto [FStartMatchBackfillRequest](#) que fornece as seguintes informações:

- ID do tíquete a ser atribuído à solicitação de alocação. Essa informação é opcional. Caso nenhum ID seja fornecido, o Amazon GameLift gerará um automaticamente.
- O marcador de jogos para o qual a solicitação é enviada. O ARN completo da configuração é necessário. Esse valor pode ser obtido dos dados do marcador da sessão do jogo.
- ID da sessão de jogo que está sendo alocada.
- Dados disponíveis de marcação para os jogadores atuais da sessão do jogo.

Obrigatório: Sim

Valor de retorno

Se bem-sucedido, retorna o tíquete de alocação de correspondência como um objeto `FGameLiftStringOutcome`. Se não for bem-sucedido, retornará uma mensagem de erro. O status do tíquete pode ser acompanhado usando a ação do SDK da AWS [DescribeMatchmaking\(\)](#).

StopMatchBackfill()

Cancela uma solicitação de alocação de correspondência ativa criada com [StartMatchBackfill\(\)](#). Consulte também a ação [StopMatchmaking\(\)](#) do SDK da AWS. Saiba mais sobre o [recurso de alocação do FlexMatch](#).

Sintaxe

```
FGameLiftGenericOutcome StopMatchBackfill (FStopMatchBackfillRequest  
&stopBackfillRequest);
```

Parâmetros

StopMatchBackfillRequest

Um objeto [FStopMatchBackfillRequest](#) que identifica o tíquete de marcação de jogos para cancelar:

- ID do tíquete atribuído à solicitação de alocação sendo cancelada
- o marcador de jogo que recebeu a solicitação de alocação

- a sessão do jogo associada à solicitação de alocação

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

TerminateGameSession()

Esse método está obsoleto com a versão 4.0.1. Em vez disso, o processo do servidor deve ser chamado [ProcessEnding\(\)](#) após o término de uma sessão de jogo.

Notifica o serviço do Amazon GameLift de que o processo do servidor encerrou a sessão atual do jogo. Essa ação é chamada quando o processo do servidor permanece ativo e pronto para hospedar uma nova sessão de jogo. Ele deve ser chamado somente após a conclusão do procedimento de encerramento da sessão de jogo, pois indica ao Amazon GameLift que o processo do servidor está imediatamente disponível para hospedar uma nova sessão de jogo.

Essa ação não será chamada se o processo do servidor for encerrado após o término da sessão do jogo. Em vez disso, chame [ProcessEnding\(\)](#) para sinalizar que tanto a sessão do jogo quanto o processo do servidor estão terminando.

Sintaxe

```
FGameLiftGenericOutcome TerminateGameSession()
```

Parâmetros

Essa ação não tem um parâmetro.

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

UpdatePlayerSessionCreationPolicy()

Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. Atualiza a capacidade da sessão do jogo atual para aceitar novas sessões de jogador. (Consulte também a ação [UpdateGameSession\(\)](#) na Referência de API do serviço do Amazon GameLift).

Sintaxe

```
FGameLiftGenericOutcome UpdatePlayerSessionCreationPolicy(EPlayerSessionCreationPolicy policy)
```

Parâmetros

Política

Valor indicando se a sessão do jogo aceita novos jogadores.

Tipo: enumeração `EPlayerSessionCreationPolicy`. Os valores válidos são:

- `ACCEPT_ALL` – Aceite todas as novas sessões de jogador.
- `DENY_ALL` – Recuse todas as novas sessões de jogador.

Obrigatório: Sim

Valor de retorno

Retorna um resultado genérico que consiste em sucesso ou falha com uma mensagem de erro.

Referência do SDK de servidor do Amazon GameLift para Unreal Engine: tipos de dados

Esta referência de Server SDK do Amazon GameLift pode ajudar a preparar os projetos de jogo Unreal Engine a serem usados com o Amazon GameLift. Para obter detalhes sobre o processo de integração, consulte [Adicione o Amazon GameLift ao servidor de jogos](#).

Esta API é definida em `GameLiftServerSDK.h` e `GameLiftServerSDKModels.h`.

Para configurar o plug-in do Unreal Engine e ver exemplos de código [Integre a Amazon GameLift em um projeto Unreal Engine](#).

- [Ações](#)
- Tipos de dados

FDescribePlayerSessionsRequest

Esse tipo de dados é usado para especificar quais sessões de jogador recuperar. Você pode usá-lo da seguinte forma:

- Forneça um `PlayerSessionId` para solicitar uma sessão de jogador específica.
- Forneça um `GameSessionId` para solicitar todas as sessões de jogador na sessão do jogo especificada.
- Forneça um `PlayerId` para solicitar todas as sessões de jogador para o jogador especificado.

Para grandes coleções de sessões de jogador, use os parâmetros de paginação para recuperar resultados em blocos sequenciais.

Índice

GameSessionId

Identificador da sessão de jogo exclusivo. Use esse parâmetro para solicitar todas as sessões de jogador da sessão de jogo especificada. O formato do ID da sessão de jogo é o seguinte: `arn:aws:gamelift:<region>::gamesession/fleet-<fleet ID>/<ID string>`. O valor de `<ID string>` é uma string de ID personalizada ou (se houver uma especificada quando a sessão de jogo foi criada) uma string gerada.

Tipo: sequência

Obrigatório: não

Limite

O número máximo de resultados a serem retornados. Use esse parâmetro com `NextToken` para obter resultados como um conjunto de páginas sequenciais. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: inteiro

Obrigatório: não

NextToken

Token indicando o início da próxima página sequencial de resultados. Use o token retornado com uma chamada anterior para essa ação. Para especificar o início do conjunto de resultados, não especifique um valor. Se um ID de sessão de jogador for especificado, esse parâmetro será ignorado.

Tipo: sequência

Obrigatório: não

PlayerId

Identificador exclusivo para um jogador. Os IDs de jogador são definidos pelo desenvolvedor. Consulte [Gerar IDs de jogador](#).

Tipo: sequência

Obrigatório: não

PlayerSessionId

Identificador exclusivo para uma sessão de jogador.

Tipo: sequência

Obrigatório: não

PlayerSessionStatusFilter

Status da sessão de jogador para filtrar resultados. Entre os status da sessão de jogador possíveis estão os seguintes:

- RESERVED – A solicitação de sessão de jogador foi recebida, mas o jogador ainda não se conectou ao processo de servidor e/ou foi validado.
- ACTIVE – O jogador foi validado pelo processo de servidor e está conectado no momento.
- COMPLETED – A conexão do jogador foi interrompida.
- TIMEDOUT – Uma solicitação de sessão de jogador foi recebida, mas o jogador não se conectou e/ou não foi validado dentro do tempo limite (60 segundos).

Tipo: sequência

Obrigatório: não

FProcessParameters

Este tipo de dados contém o conjunto de parâmetros enviado para o serviço do Amazon GameLift em uma chamada [ProcessReady\(\)](#).

Índice

porta

Número da porta em que o processo de servidor escutará novas conexões de jogador. O valor deve estar dentro do intervalo de portas configurado para qualquer frota que implanta essa

compilação do servidor de jogo. Esse número de porta está incluído nos objetos das sessões de jogo e jogador, que as sessões de jogo usam ao se conectar a um processo de servidor.

Tipo: inteiro

Obrigatório: Sim

logParameters

Objeto com uma lista de caminhos de diretório para arquivos de log da sessão de jogo.

Tipo: TArray<FString>

Obrigatório: não

onStartGameSession

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para ativar uma nova sessão de jogo. O Amazon GameLift chama essa função em resposta à solicitação do cliente [CreateGameSession](#). A função de retorno de chamada utiliza um objeto [GameSession](#) (definido na Referência de API do serviço do Amazon GameLift).

Tipo: FOnStartGameSession

Obrigatório: Sim

onProcessTerminate

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para forçar o desligamento do processo de servidor. Depois de chamar essa função, o Amazon GameLift aguardará cinco minutos para o processo de servidor desligar e responder com uma chamada [ProcessEnding\(\)](#) antes do desligamento do processo de servidor.

Tipo: FSimpleDelegate

Obrigatório: não

onHealthCheck

Nome da função de retorno que o serviço do Amazon GameLift invoca para solicitar um relatório de status de integridade do processo de servidor. O Amazon GameLift chama essa função a cada 60 segundos. Depois de chamar essa função, o Amazon GameLift aguardará 60 segundos por uma resposta e, se nenhuma for recebida, registra o processo do servidor como não íntegro.

Tipo: FOnHealthCheck

Obrigatório: não

onUpdateGameSession

Nome da função de retorno de chamada que o serviço do Amazon GameLift invoca para passar um objeto de sessão de jogo atualizado para o processo do servidor. O Amazon GameLift chama essa função quando uma solicitação de [preenchimento de correspondência](#) é processada para fornecer dados atualizados do marcador de jogos. Ele passa um objeto [GameSession](#), uma atualização de status (updateReason) e o ID do tíquete de alocação de correspondência.

Tipo: FOnUpdateGameSession

Obrigatório: não

FStartMatchBackfillRequest

Esse tipo de dados é usado para enviar uma solicitação de alocação de marcação de jogos. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StartMatchBackfill\(\)](#).

Índice

GameSessionArn

Identificador da sessão de jogo exclusivo. A ação de API [GetGameSessionId\(\)](#) retorna o identificador no formato de ARN.

Tipo: FString

Obrigatório: Sim

MatchmakingConfigurationArn

O identificador exclusivo, no formato de ARN, para o marcador de jogos a ser usado para essa solicitação. Para encontrar o marcador de jogos usado para criar a sessão de jogo original, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Saiba mais sobre os dados de marcação de jogos em [Trabalhar com dados de marcador de jogos](#).

Tipo: FString

Obrigatório: Sim

Jogadores

Um conjunto de dados que representa todos os jogadores que estão na sessão do jogo no momento. O marcador de jogos usa essas informações para pesquisar novos jogadores que sejam correspondências ideais para os jogadores atuais. Consulte o Guia de Referência de API do Amazon GameLift para obter uma descrição do formato do objeto Player. Para encontrar atributos, IDs e atribuições de equipe dos jogadores, procure no objeto de sessão do jogo, na propriedade de dados do marcador de jogos. Se o marcador de jogos usa latência, obtenha a latência atualizada para a região atual e a inclua nos dados de cada jogador.

Tipo: TArray<[FPlayer](#)>

Obrigatório: Sim

TicketId

O identificador exclusivo de um tíquete de solicitação de marcação de jogo ou de alocação de correspondência. Se nenhum valor for fornecido aqui, o Amazon GameLift gerará um no formato de UUID. Use esse identificador para acompanhar o status do tíquete de alocação de correspondência ou cancelar a solicitação, se necessário.

Tipo: FString

Obrigatório: não

FStopMatchBackfillRequest

Esse tipo de dados é usado para cancelar uma solicitação de alocação de marcação de jogo. As informações são fornecidas ao serviço do Amazon GameLift em uma chamada [StopMatchBackfill\(\)](#).

Índice

GameSessionArn

O identificador de sessão de jogo exclusivo associado à solicitação que está sendo cancelada.

Tipo: FString

Obrigatório: Sim

MatchmakingConfigurationArn

O identificador exclusivo do marcador de jogos para o qual essa solicitação foi enviada.

Tipo: FString

Obrigatório: Sim

TicketId

O identificador exclusivo do tíquete de solicitação de alocação a ser cancelado.

Tipo: FString

Obrigatório: Sim

Eventos de posicionamento de sessões de jogo

A Amazon GameLift emite eventos para cada solicitação de colocação de sessão de jogo à medida que ela é processada. Você pode publicar esses eventos em um tópico do Amazon SNS, conforme descrito em [Configurar notificações de eventos para o posicionamento de sessões de jogo](#). Esses eventos também são transmitidos para a Amazon CloudWatch Events quase em tempo real e com base no melhor esforço.

Este tópico descreve a estrutura dos eventos de posicionamento de sessões de jogo e fornece um exemplo para cada tipo de evento. Para obter mais informações sobre o status das solicitações de colocação de sessões de jogos, consulte [GameSessionPlacement](#) Amazon GameLift API Reference.

Sintaxe do evento de posicionamento

Os eventos são representados como objetos JSON. A estrutura do evento está em conformidade com o padrão CloudWatch Eventos, com campos de nível superior semelhantes e detalhes específicos do serviço.

Os campos de nível superior incluem o seguinte (consulte o [padrão do evento](#) para obter mais detalhes):

versão

Esse campo é sempre definido como 0 (zero).

id

Identificador de rastreamento exclusivo do evento.

detail-type

O valor é sempre `GameLift Queue Placement Event`.

origem

O valor é sempre `aws.gamelift`.

conta

A AWS conta que está sendo usada para gerenciar a Amazon GameLift.

horário

Carimbo de data/hora do evento.

região

A AWS região em que a solicitação de colocação está sendo processada. Essa é a região em que a fila de sessões de jogos em uso reside.

resources

Valor ARN da fila da sessão de jogo que está processando a solicitação de posicionamento.

PlacementFulfilled

A solicitação de posicionamento foi atendida com sucesso. Uma nova sessão de jogo foi iniciada e novas sessões de jogadores foram criadas para cada jogador listado na solicitação de posicionamento de sessões de jogo. As informações de conexão do jogador estão disponíveis.

Sintaxe detalhada:

placementId

Identificador exclusivo atribuído à solicitação de posicionamento de sessões de jogo.

porta

Número da porta da nova sessão de jogos.

gameSessionArn

Identificador ARN da nova sessão de jogos.

ipAddress

Endereço IP da sessão de jogos.

dnsName

Identificador DNS atribuído à instância que está executando a nova sessão de jogos. O formato do valor é diferente dependendo se a instância que executa a sessão de jogos está habilitada para TLS. Ao se conectar a uma sessão de jogos em uma frota habilitada para TLS, os jogadores devem usar o nome DNS e não o endereço IP.

Frotas habilitadas para TLS: <unique identifier>.<region identifier>.amazongamelift.com.

Frotas não habilitadas para TLS: ec2-<unique identifier>.compute.amazonaws.com.

startTime

Carimbo de data/hora indicando quando essa solicitação foi posicionada na fila.

endTime

Carimbo de data/hora indicando quando essa solicitação foi atendida.

gameSessionRegion

AWS Região da frota que está hospedando a sessão do jogo. Isso corresponde ao token de região noGameSessionArn.

placedPlayerSessions

Coleção de sessões de jogadores que foram criadas para cada jogador na solicitação de posicionamento de sessões de jogo.

Exemplo

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2021-03-01T15:50:52Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
  ],
  "detail": {
```

```
"type": "PlacementFulfilled",
"placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
"port": "6262",
"gameSessionArn": "arn:aws:gamelift:us-west-2::gamesession/
fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa/4444dddd-55ee-66ff-77aa-8888bbbb99cc",
"ipAddress": "98.987.98.987",
"dnsName": "ec2-12-345-67-890.us-west-2.compute.amazonaws.com",
"startTime": "2021-03-01T15:50:49.741Z",
"endTime": "2021-03-01T15:50:52.084Z",
"gameSessionRegion": "us-west-2",
"placedPlayerSessions": [
  {
    "playerId": "player-1"
    "playerSessionId": "psess-1232131232324124123123"
  }
]
}
}
```

PlacementCancelled

A solicitação de colocação foi cancelada com uma ligação para o GameLift serviço [StopGameSessionPlacement](#).

Detalhe:

placementId

Identificador exclusivo atribuído à solicitação de posicionamento de sessões de jogo.

startTime

Carimbo de data/hora indicando quando essa solicitação foi posicionada na fila.

endTime

Carimbo de data/hora indicando quando essa solicitação foi cancelada.

Exemplo

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
```

```
"detail-type": "GameLift Queue Placement Event",
"source": "aws.gamelift",
"account": "123456789012",
"time": "2021-03-01T15:50:52Z",
"region": "us-east-1",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
],
"detail": {
  "type": "PlacementCancelled",
  "placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
  "startTime": "2021-03-01T15:50:49.741Z",
  "endTime": "2021-03-01T15:50:52.084Z"
}
}
```

PlacementTimedOut

O posicionamento de sessões de jogo não foi concluído com êxito antes que o limite de tempo da fila expirasse. A solicitação de posicionamento pode ser reenviada conforme necessário.

Detalhe:

`placementId`

Identificador exclusivo atribuído à solicitação de posicionamento de sessões de jogo.

`startTime`

Carimbo de data/hora indicando quando essa solicitação foi posicionada na fila.

`endTime`

Carimbo de data/hora indicando quando essa solicitação foi cancelada.

Exemplo

```
{
  "version": "0",
  "id": "1111aaaa-bb22-cc33-dd44-5555eeee66ff",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
```



```
"account": "123456789012",
"time": "2021-03-01T15:50:52Z",
"region": "us-east-1",
"resources": [
  "arn:aws:gamelift:us-west-2:123456789012:gamesessionqueue/MegaFrogRace-NA"
],
"detail": {

  "type": "PlacementTimedOut",
  "placementId": "9999ffff-88ee-77dd-66cc-5555bb44aa",
  "startTime": "2021-03-01T15:50:49.741Z",
  "endTime": "2021-03-01T15:50:52.084Z"
}
}
```

PlacementFailed

A Amazon não GameLift conseguiu atender à solicitação de sessão do jogo. Isso geralmente é causado por um erro interno inesperado. A solicitação de posicionamento pode ser reenviada conforme necessário.

Detalhe:

placementId

Identificador exclusivo atribuído à solicitação de posicionamento de sessões de jogo.

startTime

Carimbo de data/hora indicando quando essa solicitação foi posicionada na fila.

endTime

Carimbo de data/hora indicando quando essa solicitação falhou.

Exemplo

```
{
  "version": "0",
  "id": "39c978f3-ba46-3f7c-e787-55bfcca1bd31",
  "detail-type": "GameLift Queue Placement Event",
  "source": "aws.gamelift",
  "account": "252386620677",
```

```
"time": "2021-03-01T15:50:52Z",
"region": "us-east-1",
"resources": [
  "arn:aws:gamelift:us-west-2:252386620677:gamesessionqueue/MegaFrogRace-NA"
],
"detail": {
  "type": "PlacementFailed",
  "placementId": "e4a1119a-39af-45cf-a990-ef150fe0d453",
  "startTime": "2021-03-01T15:50:49.741Z",
  "endTime": "2021-03-01T15:50:52.084Z"
}
}
```

Gerando estimativas de preços do Amazon GameLift

Com AWS Pricing Calculator, é possível [criar uma estimativa de preço para o Amazon GameLift](#). Você não precisa de uma Conta da AWS ou de conhecimento aprofundado da AWS para usar a calculadora.

A calculadora AWS Pricing Calculator orienta você nas decisões que afetam os custos do serviço para dar uma ideia de quanto o Amazon GameLift pode custar para seu projeto de jogo. Caso ainda não tenha certeza de como planeja usar o Amazon GameLift, use os valores padrão para gerar uma estimativa. Ao planejar o uso da produção, a calculadora pode ajudá-lo a testar possíveis cenários e gerar estimativas mais precisas.

Use AWS Pricing Calculator para gerar estimativas para as seguintes opções de hospedagem do Amazon GameLift:

- [Fazer a estimativa da hospedagem do Amazon GameLift](#)
- [Estime o FlexMatch autônomo do Amazon GameLift](#)

Fazer a estimativa da hospedagem do Amazon GameLift

Essa opção fornece uma estimativa de custo para hospedar seus jogos nos servidores gerenciados do Amazon GameLift, incluindo os custos de uso da instância do servidor e transferência de dados. A marcação de jogos do FlexMatch está incluído no custo da hospedagem gerenciada do Amazon GameLift.

Caso esteja hospedando ou planeja hospedar servidores de jogos em mais de uma região da AWS ou em mais de um tipo de instância, crie uma estimativa para cada região e tipo de instância.

Instâncias do Amazon GameLift

Esta seção ajuda você a estimar o tipo e o número de recursos de computação necessários para hospedar sessões de jogo para os jogadores. O Amazon GameLift usa as [instâncias do Amazon Elastic Compute Cloud \(Amazon EC2\)](#) para gerenciar servidores de jogos. No Amazon GameLift, você implanta uma frota de instâncias com um tipo de instância e sistema operacional específicos. Caso tenha ou planeja ter várias frota, crie uma estimativa para cada frota.

Para começar, abra a página [Configurar o Amazon GameLift](#) do AWS Pricing Calculator. Adicione uma Descrição, escolha uma Região e, em seguida, escolha Estimar a hospedagem do Amazon

GameLift (Instância + Transferência de dados de saída). Em Instâncias do Amazon GameLift, preencha os seguintes campos:

- Pico de jogadores simultâneos (pico de CCU)

Esse é o número máximo de jogadores que podem se conectar aos seus servidores de jogos ao mesmo tempo. Esse campo indica quanta capacidade de hospedagem o Amazon GameLift precisa para atender à demanda máxima dos jogadores. Insira o número máximo diário de jogadores que você espera hospedar usando instâncias na região da AWS escolhida.

Por exemplo, se você quiser permitir que 1.000 jogadores se conectem ao seu jogo ao mesmo tempo, mantenha o valor padrão de **1000**.

- CCU média por hora como porcentagem do pico diário de CCU

Esse é o número médio de jogadores simultâneos por hora em um período de 24 horas. Usamos esse valor para estimar a quantidade de capacidade de hospedagem sustentada que o Amazon GameLift precisa manter para seus jogadores. Caso não tenha certeza de qual valor percentual usar, mantenha o valor padrão de **50** por cento. Para jogos com demanda estável de jogadores, é recomendável inserir um valor de **70** por cento.

Por exemplo, se seu jogo tiver uma CCU média horária de 6.000 e uma CCU máxima de 10.000, insira o valor de **60** por cento.

- Sessões de jogo por instância

Esse é o número de sessões de jogo que cada uma das instâncias do seu servidor de jogos pode hospedar simultaneamente. Entre os fatores que podem afetar esse número estão as necessidades de recursos do servidor de jogos, o número de jogadores hospedados em cada sessão do jogo e as expectativas de desempenho do jogador. Caso saiba o número de sessões de jogo simultâneas do seu jogo, insira esse valor. Como alternativa, mantenha o valor padrão **20**.

- Jogadores por sessão de jogo

Esse é o número médio de jogadores que se conectam a uma sessão de jogo, conforme definido no design do jogo. Caso tenha modos de jogo com um número diferente de jogadores, estime o número médio de jogadores por sessão de jogo em todo o jogo. O valor padrão é **8**.

- % do buffer de inatividade da instância

Essa é a porcentagem da capacidade de hospedagem não utilizada a ser mantida em reserva para lidar com picos repentinos na demanda dos jogadores. O tamanho do buffer é uma porcentagem do número total de instâncias em uma Frota. O valor padrão **10** por cento.

Por exemplo, com um buffer ocioso de 20%, uma frota que suporta jogadores com 100 instâncias ativas mantém 20 instâncias ociosas.

- Instância spot %

As frotas do Amazon GameLift podem usar uma combinação de instâncias sob demanda e instâncias spot. Enquanto as instâncias sob demanda oferecem uma disponibilidade mais confiável, as instâncias spot oferecem uma alternativa altamente econômica. É recomendável usar uma combinação para otimizar a economia e a disponibilidade. Para obter informações sobre como o Amazon GameLift usa instâncias spot, consulte [Instâncias sob demanda versus instâncias spot](#).

Nesse campo, insira a porcentagem de instâncias spot a serem mantidas em uma frota. É recomendável uma porcentagem de instância spot entre 50 e 85 por cento. O valor padrão **50** por cento.

Por exemplo, se você implantar uma frota com 100 instâncias e especificar **40** por cento, o Amazon GameLift trabalha para manter 60 instâncias sob demanda e 40 instâncias spot.

- Tipo de instância

As frotas do Amazon GameLift podem usar uma variedade de tipos de instância do Amazon EC2 que variam em capacidade de computação, memória, armazenamento e recursos de rede. Ao configurar uma frota do Amazon GameLift, escolha um tipo de instância que melhor atenda às necessidades do seu jogo. Para obter informações sobre como selecionar um tipo de instância com o Amazon GameLift, consulte [Escolhendo os GameLift recursos computacionais da Amazon](#).

Caso saiba o tipo de instância que está usando ou planeja usar na sua frota do Amazon GameLift, escolha esse tipo. Caso não tenha certeza de qual tipo escolher, considere escolher c5.large. Esse é um tipo de alta disponibilidade com tamanho e recursos médios.

- Sistema operacional

Esse campo especifica o sistema operacional no qual seus servidores de jogos são executados no Linux ou Windows. O valor padrão é Linux.

DTO (transferência de dados de saída)

Esta seção ajuda a fazer a estimativa do custo do tráfego entre seus clientes de jogos e servidores de jogos. As taxas de transferência de dados se aplicam somente ao tráfego de saída. A transferência de dados de entrada não tem custo.

Na [página Configurar o Amazon GameLift](#) do AWS Pricing Calculator, expanda o DTO (Transferência de dados) e, em seguida, preencha os seguintes campos:

- Tipo de estimativa de DTO

Escolha estimar o DTO de uma das duas maneiras a seguir, dependendo de como você acompanha a transferência de dados do seu jogo.

- Por mês (em GB) – Caso monitore o tráfego mensal dos seus servidores de jogos, escolha esse tipo.
- Por jogador – Caso acompanhe a transferência de dados por jogador, escolha esse tipo. Esse é o tipo padrão.

No campo a seguir, você estima o DTO por jogador com base no número de horas de jogador calculado na seção anterior.

- DTO por mês (em GB)

Se você escolheu o tipo de estimativa de DTO por mês (em GB), insira seu uso mensal estimado de DTO em GB de cada instância, por região.

- DTO por jogador

Se você escolheu o tipo de estimativa de DTO por jogador, insira o uso estimado de DTO do seu jogo por jogador em KB/s. O valor padrão é **4**.

Ao terminar de configurar sua estimativa de preço do Amazon GameLift, escolha Adicionar à minha estimativa. Para obter mais informações sobre como criar e gerenciar estimativas em AWS Pricing Calculator, consulte [Criar uma estimativa, configurar um serviço e adicionar mais serviços](#) no Guia do usuário do AWS Pricing Calculator.

Estime o FlexMatch autônomo do Amazon GameLift

Esta opção fornece uma estimativa de custo para usar a marcação de jogos do FlexMatch como um serviço independente enquanto hospeda seus jogos com outra solução de servidor de jogos. Isso

inclui hospedagem autogerenciada do Amazon GameLift com FleetIQ e hospedagem on-premises, ponto a ponto ou tipos de dados primitivos de computação em nuvem. Os custos autônomos do FlexMatch são baseados na potência computacional usada.

Caso tenha ou planeja ter mais de um marcador de jogos em diferentes regiões da AWS, crie uma estimativa para cada região.

Note

O Amazon GameLift FlexMatch está disponível nas seguintes regiões: Leste dos EUA (Norte da Virgínia), Oeste dos EUA (Oregon), Ásia-Pacífico (Seul), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), Europa (Frankfurt), Europa (Irlanda).

Para começar, abra a página [Configurar o Amazon GameLift](#) do AWS Pricing Calculator. Adicione uma Descrição, escolha uma Região e, em seguida, escolha Estimar o FlexMatch autônomo do Amazon GameLift. Em Amazon GameLift FlexMatch, preencha os seguintes campos:

- Pico de jogadores simultâneos (pico de CCU)

Esse é o número máximo de jogadores que podem se conectar aos seus servidores de jogo ao mesmo tempo e solicitar a marcação de jogos. Insira o número máximo diário de jogadores que você espera igualar em sessões de jogo na região escolhida.

Por exemplo, se você quiser combinar até 1.000 jogadores por vez, mantenha o valor padrão de **1000**.

- CCU média por hora como porcentagem do pico diário de CCU

Esse é o número médio de jogadores simultâneos por hora em um período de 24 horas. Esse valor ajuda a estimar o volume de suas solicitações de marcação de jogos. Caso não tenha certeza de qual valor percentual usar, mantenha o valor padrão de **50** por cento. Para jogos com demanda estável de jogadores, é recomendável inserir um valor **70** por cento.

Por exemplo, se seu jogo tiver uma CCU média horária de 6.000 e uma CCU máxima de 10.000, insira o valor de **60** por cento.

- Número de jogadores por partida

Esse é o número médio de jogadores que competem em uma sessão de jogo, conforme definido no design do seu jogo. Caso tenha modos de jogo com números diferentes de jogadores, estime o número médio de jogadores por sessão de jogo em todo o jogo. O valor padrão é **8**.

- Duração do jogo (em minutos)

Esse é o tempo médio que os jogadores permanecem em uma sessão de jogo do início ao fim. Esse valor ajuda a determinar com que frequência os jogadores podem precisar de uma nova partida. Insira uma duração média do jogo em minutos para seus jogadores. O valor padrão é **1**.

- Complexidade das regras de marcação de jogos

A complexidade das regras de marcação de jogos se refere ao número e tipo de regras usada para combinar jogadores. O nível de complexidade do conjunto de regras ajuda a determinar a quantidade de poder computacional necessária para cada partida.

- Menor complexidade – Escolha essa opção se o conjunto de regras de marcação de jogos incluir poucas regras, usar tipos de regras mais simples (como regras de comparação) e tiver regras que formam partidas bem-sucedidas com menos tentativas.
- Maior complexidade – Escolha essa opção se o conjunto de regras de combinação incluir várias regras, usar tipos de regras mais complexos (como regras de distância ou latência) e tiver regras restritivas que resultam em mais falhas e exigem mais tentativas de correspondência.

Para obter mais informações sobre complexidade de regras e preços, consulte [Amazon GameLift FlexMatch](#) na página de preços do Amazon GameLift.

Ao terminar de configurar sua estimativa de preço do Amazon GameLift FlexMatch, escolha Adicionar à minha estimativa. Para obter mais informações sobre como criar e gerenciar estimativas em AWS Pricing Calculator, consulte [Criar uma estimativa, configurar um serviço e adicionar mais serviços](#) no Guia do usuário do AWS Pricing Calculator.

Cotas e regiões suportadas

Para o Service Quotas da AWS do Amazon GameLift, consulte [Cotas do Amazon GameLift](#).

Para obter informações sobre como solicitar aumentos de cota para recursos da AWS, consulte [Iquotas de serviço da AWS](#).

Para obter uma lista das Regiões da AWS que oferecem suporte ao Amazon GameLift, consulte [Regiões do Amazon GameLift](#).

Notas GameLift de lançamento da Amazon

As notas de GameLift lançamento da Amazon fornecem detalhes sobre novos recursos, atualizações e correções relacionadas ao serviço.

Versões do SDK

As tabelas a seguir listam todos os GameLift lançamentos da Amazon com informações sobre a versão do SDK. Não há necessidade de usar SDKs comparáveis para suas integrações de servidor e cliente de jogos. No entanto, as versões anteriores de um SDK podem não oferecer suporte total aos recursos mais recentes em outro.

Para obter mais informações sobre os Amazon GameLift SDKs, consulte [Suporte ao desenvolvimento com a Amazon GameLift](#).

Para obter os GameLift SDKs da Amazon mais recentes, consulte o site de download dos [GameLiftSDKs da Amazon](#).

Versão atual

3	Seção
3	SDK
3	recurso
3	real
3	para
3	login
3	#
3	para
3	Unity
3	para
3	Unreal
3	51120251
3	u
3	osterior

Versões anteriores

Liberação de serviços	AWS SDK	Server SDK:				Clientes em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
2023-12-14	1.11.225 ou posterior	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-11-02	1.11.193 ou posterior	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-09-28	1.11.144 ou posterior	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-08-17	1.11.144 ou posterior	5.1.0	5.1.1	5.1.0	5.0.0	1.2.0
2023-07-27	1.11.111 ou posterior	5.1.0	5.1.0	5.0.2	5.0.0	1.2.0
2023-06-29	1.11.111 ou posterior		5.0.4	5.0.2	5.0.0	1.2.0

Liberação de serviços	AWS SDK	Server SDK:				Client em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
2023-06-15	1.11.87 ou posterior		5.0.4	5.0.2	5.0.0	1.2.0
2023-05-25	1.11.87 ou posterior		5.0.3	5.0.2	5.0.0	1.2.0
2023-04-20	1.11.63 ou posterior				5.0.0	1.2.0
2023-04-13	1.10.21 ou posterior				5.0.0	1.2.0
2023-02-09	1.10.21 ou posterior			3.4.0	5.0.0	1.2.0
2023-01-31	1.10.21 ou posterior			3.4.0	5.0.0	1.2.0
2022-12-01	1.10.21 ou posterior			3.4.0		1.2.0
25/08/2022	1.9.333 ou posterior		3.4.2	3.4.0		1.2.0

Liberação de serviços	AWS SDK	Server SDK:				Clientes em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
2021-10-28	1.9.133 ou posterior		3.4.2	3.4.0		1.2.0
2021-06-03	1.8.168 ou posterior		3.4.2	3.4.0		1.2.0
2021-03-23	1.8.168 ou posterior		3.4.1	3.3.3		1.1.0
2021-03-16	1.8.163 ou posterior		3.4.1	3.3.3		1.1.0
2021-02-09	1.8.139 ou posterior		3.4.1	3.3.3		1.1.0
2020-12-22	1.8.95 ou posterior		3.4.1	3.3.3		1.1.0
2020-11-24	1.8.95 ou posterior		3.4.1	3.3.2		1.1.0
2020-11-11	1.8.36 ou posterior		3.4.1	3.3.2		1.1.0
2020-09-17	1.8.36 ou posterior		3.4.1	3.3.2		1.1.0
2020-08-27	1.7.310 ou posterior		3.4.0	3.3.1		1.1.0

Liberação de serviços	AWS SDK	Server SDK:				Client em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
2020-04-16	1.7.310 ou posterior		3.4.0	3.3.1		1.1.0
02-04-2020	1.7.310 ou posterior		3.4.0			1.1.0
19-12-2019	1.7.249 ou posterior		3.4.0			1.1.0
2019-11-14	1.7.210 ou posterior		3.4.0			1.1.0
2019-10-24	1.7.210 ou posterior		3.4.0			1.1.0
2019-09-03	1.7.175 ou posterior		3.4.0			1.1.0
2019-07-09	1.7.140 ou posterior		3.3.0			1.0.0
25/04/2019	1.7.91 ou posterior		3.3.0			1.0.0
2019-03-07	1.7.65 ou posterior		3.3.0			
2019-02-07	1.7.45 ou posterior		3.3.0			

Liberação de serviços	AWS SDK	Server SDK:				Client em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
14/12/2018	1.6.20 ou posterior		3.3.0			
2018-09-27	1.6.20 ou posterior		3.2.1			
2018-06-14	1.4.47 ou posterior		3.2.1			
2018-05-10	1.4.47 ou posterior		3.2.1			
15/02/2018	1.3.58 ou posterior		3.2.1			
08/02/2018	1.3.52 ou posterior		3.2.0			
2017-09-01	1.1.43 ou posterior		3.1.7			
16/08/2017	1.1.31 ou posterior		3.1.7			
2017-05-16	1.0.122 ou posterior		3.1.5			
2017-04-11	1.0.103 ou posterior		3.1.5			

Liberação de serviços	AWS SDK	Server SDK:				Clientes em tempo real para
		Plugin C# para Unity	C++	Plugin C++ para Unreal	Go	
21/02/2017	1.0.72 ou posterior		3.1.5			
2016-11-18	1.0.31 ou posterior		3.1.0			
2016-10-13	1.0.17 ou posterior		3.1.0			
01/09/2016	0.14.9 ou posterior		3.1.0			
04/08/2016	0.12.16 ou posterior		3.0.7			

Notas de release

As notas de release a seguir estão em ordem cronológica, com as atualizações mais recentes listadas primeiro. A Amazon GameLift foi lançada pela primeira vez em 2016. Para obter notas de release anteriores às listadas aqui, consulte os links de datas de lançamento em [Versões do SDK](#).

24 de abril de 2024: Amazon GameLift lança frotas de contêineres

GameLift A Amazon agora oferece uma prévia das frotas de contêineres, que oferecem maior portabilidade, escalabilidade, tolerância a falhas e agilidade.

Em frotas de contêineres, as instâncias do Amazon EC2 hospedam um ou mais de seus contêineres. Esses contêineres incluem seu servidor de jogos junto com tudo o que for necessário, incluindo dependências e configurações. Exemplos de dependências incluem SDKs e pacotes de software.

Depois de carregar seu contêiner para seu Amazon Elastic Container Registry privado, a Amazon GameLift preenche sua frota com o contêiner.

Para funcionar em uma frota de contêineres, seu servidor de jogos deve ser executado em Linux e estar integrado ao Server SDK 5.x. Em uma frota de contêineres, você tem um controle refinado dos recursos de hospedagem para poder otimizar o consumo de recursos, como unidades de CPU e memória. Você também pode hospedar vários servidores de jogos em um contêiner para reduzir o uso de recursos.

Em uma frota de contêineres, você obtém muitos dos mesmos benefícios que outros tipos de frotas têm, como tipos de instância sob demanda, escalabilidade (automática e manual), filas e matchmaking. Você também obtém as mesmas métricas de outros tipos de frota, além de algumas novas para contêineres. As frotas de contêineres oferecem alcance global aos jogadores nos seguintes locais e regiões:

- ap-northeast-1
- ap-northeast-2
- ap-southeast-2
- eu-central-1
- eu-west-1
- us-east-1
- us-west-2

Para alcançar ainda mais regiões e zonas locais, crie frotas de contêineres com vários locais.

Saiba mais:

- [Gerenciando hospedagem com GameLift contêineres da Amazon](#), Amazon GameLift Developer Guide
- [CreateContainerGroupDefinition](#), Referência de GameLift API da Amazon

13 de fevereiro de 2024: Amazon GameLift lança melhorias nos SDKs e simplifica a instalação do GameLift plug-in da Amazon para Unreal Engine

Versões atualizadas do SDK:

- SDK do Go Server, versão 5.1.0

- SDK do servidor C#, versão 5.1.2
- SDK do servidor C++, versão 5.1.2

Fizemos as seguintes melhorias:

- Melhorou a confiabilidade do SDK adicionando reconexão automática em caso de interrupção da rede.
- [Go] Agora você pode ligar `InitSDK()` com ou sem parâmetros do servidor. Os servidores de jogos executados em frotas EC2 GameLift gerenciadas pela Amazon leem os parâmetros do servidor diretamente das variáveis de ambiente. Os servidores de jogos nas GameLift Anywhere frotas da Amazon devem ligar `InitSDK()` com os parâmetros do servidor.

Versões atualizadas do plugin:

- GameLift Plugin Amazon para Unreal Engine, versão 1.1.0
- GameLift Plugin Amazon para Unity, versão 2.1.0
- Plugin C++ Server SDK para Unreal, versão 5.1.1
- Plugin C# Server SDK para Unity, versão 5.1.2

Fizemos as seguintes melhorias:

- [GameLift Plugin da Amazon para Unreal Engine] Atualizou as instruções de instalação e simplificou o empacotamento. Esse plug-in agora inclui a versão mais recente do SDK do servidor C++ para Unreal.
- Atualizou os plug-ins para oferecer suporte à versão mais recente do SDK do GameLift servidor.

Saiba mais:

- [Integrando jogos com o GameLift plug-in da Amazon para Unreal Engine, Amazon Developer Guide GameLift](#)
- [Downloads de GameLift plug-ins e SDK da Amazon](#)

14 de dezembro de 2023: a Amazon GameLift adiciona a capacidade de atualizar as propriedades do jogo das sessões ativas do jogo

Você já conseguiu definir as propriedades do jogo ao criar sessões de jogo e pesquisar as propriedades especificadas nas sessões do jogo. Agora você também pode adicionar e atualizar essas propriedades em uma sessão de jogo ativa.

Por exemplo, seus jogadores votam em um mapa no qual querem jogar. Seu cliente de jogo liga `UpdateGameSession` para modificar um `GameProperty` valor para `{"Key": "map", "Value": "jungle"}`. Seu jogo então implementa o novo mapa para os jogadores na sessão de jogo.

Os administradores do jogo também podem recuperar dados úteis das propriedades do jogo usando a `SearchGameSessions` operação. Por exemplo, os administradores podem listar sessões de jogo que têm um `Status` valor de `ACTIVE` e esta propriedade do jogo: `{"Key": "map", "Value": "desert"}`.

Saiba mais:

- [the section called “Adicione GameLift a Amazon a um cliente de jogo”](#), Guia do GameLift desenvolvedor da Amazon
- [GameProperty](#), Referência de GameLift API da Amazon
- [UpdateGameSession](#), Referência de GameLift API da Amazon
- [SearchGameSessions](#), Referência de GameLift API da Amazon

21 de novembro de 2023: a Amazon GameLift lança suporte para ferramentas de infraestrutura como código, como Terraform e Pulumi, desenvolvidas por AWS Cloud Control API

Agora você pode gerenciar toda a sua pilha de GameLift recursos da Amazon usando as ferramentas de Infraestrutura como Código (IaC). Essas ferramentas incluem ferramentas de terceiros AWS CloudFormation, como Terraform e Pulumi. Com esse suporte adicional, agora você pode se concentrar em criar seu jogo e aproveitar DevOps estratégias para cuidar do gerenciamento de recursos, CI/CD e implantação para seus clientes.

Agora você também pode provisionar e configurar todos os tipos de GameLift recursos da Amazon usando a API AWS Cloud Control. Você pode continuar trabalhando com recursos usando as GameLift APIs da Amazon ou os AWS CloudFormation modelos da Amazon GameLift.

Para obter detalhes sobre os GameLift recursos da Amazon disponíveis por meio do IaC, consulte a [referência do tipo](#) de GameLift recurso da Amazon. GameLift

Além disso, agora você pode escalar automaticamente suas frotas usando AWS CloudFormation modelos ou a API AWS Cloud Control usando a nova propriedade [Fleet:ScalingPolicies](#).

A API Cloud Control oferece aos desenvolvedores um conjunto padrão de APIs para criar, ler, atualizar, excluir e listar recursos (CRUDL) em centenas de AWS serviços e várias ferramentas de terceiros, como Terraform e Pulumi.

Saiba mais:

- [AWS CloudFormation](#)
- [AWS API de controle de nuvem](#)
- [AWS Fornecedor CC Terraform](#)
- [Pulumi](#)

16 de novembro de 2023: Amazon GameLift atualiza plug-in autônomo para Unity

Versões atualizadas do SDK: GameLift plug-in Amazon para Unity, versão 2.0.0

O GameLift plug-in da Amazon para Unity fornece ferramentas e fluxos de trabalho que simplificam as etapas para colocar seu jogo Unity em funcionamento para hospedagem na nuvem com a Amazon. GameLift A Amazon é um serviço totalmente gerenciado que permite aos desenvolvedores de jogos gerenciar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões.

Com esta versão, o plug-in para Unity é atualizado para usar os GameLift recursos mais recentes da Amazon, incluindo o SDK do servidor versão 5.x e suporte para testes locais com o Amazon GameLift Anywhere. O plug-in é compatível com as versões Unity 2021.3 LTS e 2022.3 LTS do Unity.

Os principais recursos do plug-in incluem:

- Fluxos de trabalho de interface do usuário guiados no editor Unity para os seguintes cenários:
 - Teste sua integração de jogos com a Amazon GameLift usando sua estação de trabalho local como host. Esse fluxo de trabalho ajuda você a configurar uma GameLift Anywhere frota da Amazon para sua máquina local, iniciar instâncias do seu servidor e cliente de jogos, solicitar uma sessão de jogo pela Amazon GameLift e participar do jogo.

- Implante uma solução de hospedagem em nuvem para seu servidor de jogos integrado com EC2 GameLift gerenciado pela Amazon e AWS recursos de suporte. Esse fluxo de trabalho ajuda você a configurar seu jogo para hospedagem na nuvem e fornece três cenários de implantação:
 - Implante o servidor do jogo em uma única frota.
 - Implante o servidor do jogo em um conjunto de frotas spot de baixo custo em várias AWS regiões.
 - Implante o servidor do jogo com um FlexMatch matchmaker.
- Capacidade de configurar perfis de usuário vinculados a um usuário AWS da conta e definir uma AWS região padrão. Você pode manter vários perfis para trabalhar em diferentes AWS contas, usuários da conta e regiões.
- Conveniências especiais que ajudam a agilizar os processos de GameLift integração e implantação da Amazon, incluindo:
 - Cada solução de hospedagem inclui AWS recursos de suporte, incluindo um grupo de usuários do Amazon Cognito que fornece IDs de jogadores e validação de jogadores exclusivos. As soluções também incluem um bucket do Amazon S3 para armazenamento, notificação de eventos do Amazon SNS AWS Lambda , funções e outros recursos.
 - Para o Anywhere fluxo de trabalho, o plug-in automatiza as configurações de parâmetros do servidor necessárias.
 - Para o fluxo de trabalho do Amazon EC2, cada solução de implantação fornece um serviço de back-end de cliente integrado usando funções Lambda. O serviço de back-end fica entre o cliente do jogo e o GameLift serviço da Amazon e gerencia todas as chamadas diretas para o GameLift serviço da Amazon.
- Conteúdo para testes de integração, incluindo ativos e código para um simples exemplo de jogo multijogador para ilustrar a integração entre servidor e cliente de jogo.
- Documentação do plug-in com orientação detalhada de integração e código de amostra.

Todos os cenários de implantação, inclusive para frotas do Amazon EC2 Anywhere e do Amazon, usam AWS CloudFormation modelos para descrever e implantar os AWS recursos para a solução do seu jogo. Esses modelos estão incluídos no download do GameLift plug-in da Amazon. Você pode usá-los como estão ou personalizá-los para o seu jogo.

Saiba mais:

- [Guia do GameLift plugin Amazon para Unity para servidor SDK 5.x](#), Guia do GameLift desenvolvedor da Amazon
- [Baixe o plugin em GitHub](#)
- [Sobre a GameLift hospedagem na Amazon](#)
- [GameLift Fórum da Amazon](#)

2 de novembro de 2023: Amazon GameLift adiciona suporte para credenciais compartilhadas

Versões atualizadas do SDK: AWS SDK 1.11.193

O novo recurso de credenciais GameLift compartilhadas da Amazon permite que aplicativos implantados em frotas gerenciadas do EC2 interajam com outros recursos. AWS Essa atualização afeta os aplicativos que você agrupa e implanta junto com os binários do servidor de jogos integrados ao SDK do servidor versão 5.x ou posterior. (Os executáveis do servidor de jogos já podem solicitar credenciais usando a ação `GetFleetRoleCredentials()` 5.x do SDK de servidor.)

Por exemplo, se você quiser implantar sua compilação de servidor de jogos com um CloudWatch agente da Amazon para coletar métricas de instância do EC2 e outros dados, o agente precisa de permissão para interagir com seus CloudWatch recursos. Para fazer isso, você deve primeiro configurar uma função (AWS Identity and Access Management IAM) com permissões para usar os CloudWatch recursos e, em seguida, configurar uma frota com a função do IAM e as credenciais compartilhadas ativadas. Quando a Amazon GameLift implanta sua versão de servidor de jogos em cada instância do EC2, ela gera um arquivo de credenciais compartilhado e o armazena na instância. Todos os aplicativos na instância podem usar as credenciais compartilhadas. A Amazon atualiza GameLift automaticamente as credenciais temporárias durante toda a vida útil da instância.

Você pode ativar as credenciais compartilhadas ao criar uma frota EC2 gerenciada usando os seguintes métodos:

- No fluxo de trabalho de criação de frotas do GameLift console Amazon.
- Ao chamar a operação GameLift de API do serviço Amazon CreateFleet usando o novo parâmetro `InstanceRoleCredentialsProvider`.

- Ao chamar a operação AWS CLI `aws gamelift create-fleet` com o parâmetro `instance-role-credentials-provider`

Saiba mais:

- [Comuniquese com outros AWS recursos de suas frotas](#), Amazon GameLift Developer Guide
- [CreateFleet, InstanceRoleCredentialsProvider](#), Referência de GameLift API da Amazon
- [Configure uma função de serviço do IAM](#), Amazon GameLift Developer Guide

28 de setembro de 2023: Amazon GameLift lança novo plug-in autônomo para o Unreal Engine

Versões atualizadas do SDK: GameLift plug-in Amazon para Unreal Engine versão 1.0.0

O GameLift plug-in da Amazon para Unreal Engine fornece ferramentas e fluxos de trabalho que simplificam suas etapas para colocar um jogo em funcionamento com a Amazon GameLift para hospedagem na nuvem. GameLift A Amazon é um serviço totalmente gerenciado que permite aos desenvolvedores de jogos gerenciar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões. O plug-in suporta as versões 5.0, 5.1 e 5.2 da UE. Os principais recursos incluem:

- Fluxos de trabalho de UI guiados no editor Unreal percorrem os seguintes caminhos:
 - Teste sua integração de jogos com a Amazon GameLift usando sua estação de trabalho local como host. Esse fluxo de trabalho ajuda você a configurar uma GameLift Anywhere frota da Amazon para sua máquina local, iniciar instâncias do seu servidor e cliente de jogos, solicitar uma sessão de jogo pela Amazon GameLift e obter informações de conexão para a nova sessão de jogo.
 - Implante uma solução de hospedagem em nuvem do Amazon EC2 para seu servidor de jogos integrado. Esse fluxo de trabalho ajuda você a configurar seu jogo para hospedagem na nuvem e fornece três cenários de implantação diferentes: implante em uma única frota, implante em um conjunto de frotas spot em várias regiões ou implante em um conjunto de frotas com um FlexMatch matchmaker. A solução para cada cenário de implantação inclui GameLift recursos e recursos de suporte AWS da Amazon.
- Capacidade de configurar perfis de usuário vinculados a um usuário AWS da conta e definir uma AWS região padrão. Você pode manter vários perfis para trabalhar em diferentes AWS contas, usuários da conta e regiões.

- Conveniências especiais que ajudam a agilizar os processos de GameLift integração e implantação da Amazon, incluindo:
 - Cada solução de hospedagem inclui AWS recursos de suporte, incluindo um grupo de usuários básico do Amazon Cognito que fornece IDs de jogadores exclusivos, um bucket do Amazon S3 para armazenamento, notificação de eventos e funções do Amazon SNS. AWS Lambda
 - Para o Anywhere fluxo de trabalho, o plug-in automatiza as configurações de parâmetros do servidor necessárias usando argumentos de linha de comando.
 - Para o fluxo de trabalho do Amazon EC2, cada solução de implantação fornece um serviço de back-end de cliente integrado usando funções Lambda. O serviço de back-end recebe solicitações de clientes de jogos e as repassa para o GameLift serviço da Amazon.
- Conteúdo para testes de integração, incluindo um mapa inicial do jogo e dois mapas de teste com esquemas básicos e elementos de interface do usuário.
- Documentação do plug-in com orientação detalhada de integração e código de amostra.

Todos os cenários de implantação, inclusive para frotas do Amazon EC2 Anywhere e do Amazon, usam AWS CloudFormation modelos para descrever as soluções. O plug-in usa esses modelos ao implantar GameLift recursos da Amazon para seu jogo. Esses modelos estão incluídos no download do GameLift plug-in da Amazon e são editáveis. Você pode usá-las como estão ou modificá-las para o seu jogo.

Saiba mais:

- [Integrando jogos com o GameLift plugin Amazon para Unreal Engine](#), Guia do GameLift desenvolvedor da Amazon
- [Baixe o plugin em GitHub](#)
- [Sobre a GameLift hospedagem na Amazon](#)
- [GameLift Fórum da Amazon](#)

17 de agosto de 2023: Amazon GameLift oferece hospedagem de servidores de jogos com processadores AWS Graviton

Versões atualizadas do SDK: AWS SDK 1.11.144

Com a Amazon, agora GameLift você pode hospedar seus jogos na nuvem usando instâncias EC2 com processadores AWS Graviton. Projetadas AWS com processadores baseados em ARM64,

as instâncias Graviton oferecem a melhor relação preço/desempenho para cargas de trabalho em nuvem usando o EC2, com até 40% de melhoria em relação às instâncias comparáveis baseadas em x86. Os processadores Graviton3 mais recentes oferecem desempenho computacional até 25% melhor em relação às versões anteriores.

Com a Amazon GameLift, agora você pode selecionar entre essas novas instâncias da família AWS Graviton:

- Instâncias baseadas em Graviton2: c6g, c6gn, r6g, m6g, g5g
- Instâncias baseadas em Graviton3: c7g, r7g, m7g

Saiba mais:

- [AWS Processador Graviton](#): saiba mais sobre os benefícios e os usos práticos das instâncias EC2 baseadas em Graviton.
- [Introdução ao Graviton](#): tenha uma visão geral das instâncias baseadas em Graviton e insights sobre como os aplicativos são executados nelas, dependendo do sistema operacional, dos idiomas e dos tempos de execução.

Note

As instâncias do Graviton Arm exigem um GameLift servidor Amazon baseado no sistema operacional Linux. O Server SDK 5.1.1 ou mais recente é necessário para C++ e C#. O Server SDK 5.0 ou mais recente é necessário para o Go. Essas instâncias não oferecem out-of-the-box suporte para a instalação do Mono no Amazon Linux 2023 (AL2023) ou no Amazon Linux 2 (AL2).

27 de julho de 2023: a Amazon GameLift lança o SDK 5.1.0 do servidor com suporte adicional para o desenvolvimento do Unity

Versões atualizadas do SDK: SDK de servidor para C++, C#/Unity, Unreal 5.1.0

A versão mais recente do SDK do GameLift servidor Amazon fornece atualizações para C++, C# e o plug-in Unreal, além de um novo plug-in para uso com o mecanismo de jogo Unity. Os desenvolvedores de jogos integram o SDK GameLift do servidor Amazon aos servidores de jogos que eles implantam para hospedagem na Amazon GameLift.

A versão mais recente do SDK de servidor contém as seguintes atualizações, que incluem várias solicitações de clientes:

- Baixe pacotes SDK específicos do idioma — O [site de GameLift download atualizado da Amazon](#) contém pacotes de SDK para cada idioma. Você pode baixar as versões atuais ou anteriores.
- Novo plug-in de SDK de servidor C# para Unity — O novo pacote de SDK de servidor para Unity contém bibliotecas C# criadas que você pode instalar usando o gerenciador de pacotes no Unity Editor (consulte o novo guia de integração do [Unity](#)). Essas bibliotecas incluem as dependências necessárias por meio de UnityNuGet. Você pode usar esse plug-in com o Unity 2020.3 LTS, 2021.3 LTS e 2022.3 LTS para Windows e Mac OS. Ele suporta os perfis .NET Framework e .NET Standard do Unity, com o .NET Standard 2.1 e .NET 4.x.
- Solução .NET consolidada para C#: O SDK de servidor para C# agora oferece suporte a o .NET Framework 4.6.2 (atualizado da versão 4.6.1) e o .NET 6.0 em uma única solução. O .NET Standard 2.1 está disponível com as bibliotecas criadas pelo Unity.
- Atualizações do Server SDK 5.1.0
 - [C++, C#, Unreal] Agora você pode chamar `InitSDK()` com ou sem parâmetros do servidor. Os servidores de jogos executados em frotas EC2 GameLift gerenciadas pela Amazon leem os parâmetros do servidor diretamente das variáveis de ambiente. Os servidores de jogos nas GameLift Anywhere frotas da Amazon devem ligar `InitSDK()` com os parâmetros do servidor.
 - [C++, C#, Unreal] As chamadas do SDK de servidor melhoraram as mensagens de erro.
 - [SDK para C++] Para melhorar os tempos de compilação do SDK de servidor, o sinalizador de compilação `-DRUN_CLANG_FORMAT` está desativado por padrão. Você pode habilitá-lo com `-DRUN_CLANG_FORMAT=1`.
 - [SDK para C++] Ao criar as bibliotecas sem as bibliotecas padrão (`-DGAMELIFT_USE_STD=0`), `InitSDK()` não usa mais tipos de `std::` dados.
- Documentação expandida do SDK 5.x para servidor
 - Guias de referência do SDK de servidor atualizados para C++, C#/Unity e Unreal, incluindo cobertura expandida de todos os tipos de dados.
 - [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)
 - [Referência do SDK 5.x do servidor Amazon GameLift para C++](#)
 - [Referência do Server SDK 5.x do servidor do Amazon GameLift para Unreal Engine](#)
 - Novas versões dos guias de integração do SDK 5 do servidor para plug-ins Unity e Unreal
 - [Integre o Amazon GameLift em um projeto do Unity](#)
 - [Integre a Amazon GameLift em um projeto Unreal Engine](#)

- Atualizações adicionais da documentação
 - Documentação revisada para operações de API GameLift de serviços da Amazon [GetComputeAccess](#) e [GetInstanceAccess](#) para esclarecer os procedimentos de acesso remoto com base na versão do SDK GameLift do servidor Amazon em uso.
 - Descrições revisadas [GameSessionPlacement](#) para documentar como as informações da sessão de jogo são transitórias quando uma colocação está no status “pendente”.

13 de julho de 2023: Amazon GameLift adiciona métricas de hardware de frota

Agora você pode monitorar métricas de desempenho de hardware para suas frotas EC2 GameLift gerenciadas pela Amazon. As métricas incluem métricas de instância do EC2 para utilização da CPU, volume de tráfego de rede e atividade de leitura/gravação em disco. Para a Amazon GameLift, essas métricas descrevem todas as instâncias ativas em um local de frota. Você pode visualizar essas métricas de hardware da frota usando um CloudWatch painel da Amazon no AWS Management Console. Você também pode visualizá-los no GameLift console da Amazon nos detalhes da frota.

Saiba mais:

- [Monitorar o Amazon GameLift com o Amazon CloudWatch](#)(Métricas para frotas), Amazon GameLift Developer Guide

29 de junho de 2023: Amazon GameLift lança suporte para Amazon Linux 2023

Versões atualizadas do SDK: AWS SDK 1.11.111

GameLift Os clientes da Amazon agora podem usar o sistema operacional Amazon Linux 2023 para hospedar seus servidores de jogos. O AL2023 oferece várias melhorias em relação ao AL2, incluindo segurança. Esse sistema operacional está disponível em todos, Regiões da AWS com exceção das regiões da China.

Os clientes podem usar os sistemas operacionais Linux mais recentes e continuar recebendo atualizações críticas de segurança quando o suporte ao Amazon Linux (AL1) terminar em dezembro de 2023. O suporte para o Amazon Linux 2 continua até 2025.

Saiba mais:

- [Perguntas frequentes sobre o Amazon GameLift Linux Server](#)

- [Comparação entre Amazon Linux 2 e Amazon Linux 2023](#)
- Links de referência GameLift da API da Amazon:
 - [AWS Ação do SDK CreateBuild](#)
 - [Comando da CLI upload-build](#)
 - [Comando da CLI create-build](#)

25 de maio de 2023: o Amazon GameLift FleetIQ adiciona filtro para excluir posicionamentos de sessões de jogo em instâncias de drenagem

Versões atualizadas do SDK: AWS SDK 1.11.87

Se você usa o Amazon GameLift FleetIQ para hospedagem de jogos, agora você pode evitar posicionamentos de sessões de jogos em instâncias que estão se esgotando no momento. As instâncias de drenagem estão sinalizadas para encerramento, mas elas ainda podem ser selecionadas para hospedar novas sessões de jogo se nenhum outro recurso de hospedagem estiver disponível. Com esse novo atributo, você pode excluir totalmente o uso de instâncias de drenagem.

Use esse atributo ao ligar `ClaimGameServer` para encontrar servidores de jogos disponíveis. Adicione o novo `FilterOption` parâmetro e defina os status de instância permitidos somente como ATIVO. Em resposta, o Amazon GameLift FleetIQ analisa somente as instâncias ativas ao pesquisar e reivindicar um servidor de jogo disponível.

Saiba mais:

- [ClaimGameServer](#) na Amazon GameLift API Reference
- [Como o FleetIQ funciona no Amazon FleetIQ](#) Developer Guide GameLift

16 de maio de 2023: a Amazon GameLift oferece suporte à etiquetagem de alocação de custos para frotas

GameLift Os clientes da Amazon agora podem usar tags de alocação de AWS Billing custos para organizar seus custos de hospedagem de jogos. Você pode atribuir etiquetas de alocação de custos a recursos individuais da frota do Amazon GameLift EC2 para monitorar como suas frotas estão contribuindo para os custos gerais de hospedagem.

Saiba mais:

- [Gerencie os custos de hospedagem de jogos](#)
- [Uso de tags de alocação de custos AWS](#) no AWS Billing Guia do usuário.

20 de abril de 2023: Amazon GameLift lança suporte para o Windows Server 2016

Versões atualizadas do SDK: AWS SDK 1.11.63

GameLift Os clientes da Amazon agora podem usar o sistema operacional Windows Server 2016 para hospedar seus servidores de jogos. Esse sistema operacional está disponível em todos as Regiões da AWS. Os clientes podem usar o sistema operacional Windows mais recente e continuar recebendo atualizações críticas de segurança à medida que a Microsoft encerra seu suporte para o Windows Server 2012 em outubro de 2023.

A partir de hoje, novos clientes que precisam de um ambiente de runtime do Windows devem especificar o Windows Server 2016 ao criar novas versões de servidores de jogos para hospedagem. Os clientes existentes podem continuar criando novas versões e frotas com o Windows Server 2012, mas devem concluir a migração com o Windows Server 2016 antes da data de fim do suporte da Microsoft, em 10 de outubro de 2023.

Esta atualização inclui as seguintes alterações de serviço:

- Ao criar uma compilação de servidor de jogos usando comandos Amazon GameLift SDK ou CLI, agora você deve definir explicitamente o sistema operacional. Não há valor padrão. Para implantar seu servidor de jogos no Windows Server 2016, você pode usar o valor `WINDOWS_2016`.
- Ao criar uma compilação de servidor de jogos usando o GameLift console da Amazon, você deve selecionar um sistema operacional a partir dos valores disponíveis. Se você já é um cliente com frotas ativas do Windows Server 2012, você pode escolher entre uma `WINDOWS_2012` ou outra `WINDOWS_2016` opção.

Saiba mais:

- Links de referência GameLift da API da Amazon:
 - [Comando da CLI `upload-build`](#)
 - [Comando da CLI `create-build`](#)
 - [AWS Ação do SDK `CreateBuild`](#)

- [GameLift Perguntas frequentes da Amazon para Windows 2012](#)

13 de abril de 2023: Amazon GameLift lança servidor SDK 5.x para Unreal

Versões atualizadas do SDK: SDK 5.0.0 de servidor para Unreal

A versão mais recente do plug-in GameLift leve da Amazon para Unreal Engine agora é baseada no Amazon GameLift Server SDK 5.x. Para começar a integrar seu ambiente Unreal Engine com a Amazon, GameLift consulte os links a seguir.

Saiba mais:

- [Integre a Amazon GameLift em um projeto Unreal Engine](#)
- [Adicione o Amazon GameLift ao servidor de jogos](#)
- [Referência do SDK 5.x do servidor Amazon GameLift para C++](#)

14 de março de 2023: Amazon GameLift lança uma nova experiência de console

O novo GameLift console da Amazon inclui essas melhorias:

- Navegação aprimorada — O novo painel de navegação facilita a navegação entre os recursos da Amazon GameLift .
- Página GameLift inicial da Amazon — A nova página inicial fornece links para documentação útil, exibe uma visão geral de alto nível da Amazon GameLift e fornece suporte por meio de links para documentação, perguntas frequentes e. AWS re:Post
- CloudWatch Métricas aprimoradas da Amazon — As GameLift métricas da Amazon agora estão disponíveis no GameLift console da Amazon e em seus CloudWatch painéis. Essa atualização também inclui novas métricas de desempenho, utilização e sessões de jogadores.

Saiba mais:

- [Visualizar os dados do jogo no console](#)
- [Gerenciar recursos de hospedagem do Amazon GameLift](#)
- [Construindo um FlexMatch casamenteiro](#)

14 de fevereiro de 2023: a Amazon GameLift agora oferece suporte à criptografia do lado do servidor para tópicos do Amazon SNS

A criptografia do lado do servidor (SSE) para tópicos do SNS criptografa seus dados confidenciais em repouso. SSE usa as teclas AWS Key Management Service (AWS KMS) para proteger o conteúdo de seus tópicos do SNS.

Saiba mais:

- [Configurar notificações de eventos para o posicionamento de sessões de jogo](#)
- [FlexMatcheventos de matchmaking](#)
- [Criptografia em repouso](#)

9 de fevereiro de 2023: o Amazon GameLift Server SDK oferece suporte a o.NET 6 com C #10

Versões atualizadas do SDK: SDK 5.0.0 de servidor para .NET 6 Nenhuma atualização do SDK é necessária.

Se você usa a Unity Real-Time Development Platform, continue usando o Amazon GameLift Server SDK 5.0.0 com .NET 4.6. O Unity não é compatível com o.NET 6.

Saiba mais:

- Baixe a versão mais recente do SDK do GameLift servidor Amazon em [Amazon GameLift Getting Started](#)
- [Referência 5.x do SDK do servidor Amazon GameLift para C# e Unity](#)

31 de janeiro de 2023: o SDK GameLift do servidor Amazon oferece suporte à linguagem Go

Versões atualizadas do SDK: SDK 5.0.0 de servidor para Go

Saiba mais:

- Baixe a versão mais recente do SDK do GameLift servidor Amazon em [Amazon GameLift Getting Started](#)
- [Referência do SDK do servidor do Amazon GameLift para Go](#)

1º de dezembro de 2022: Amazon GameLift lança Amazon GameLift Anywhere e Amazon GameLift Server SDK 5.0

Versões atualizadas do SDK: AWS SDK 1.10.21, Server SDK 5.0.0 para C++ e C#

A Amazon GameLift Anywhere usa seus recursos de servidor de jogos para hospedar servidores de GameLift jogos da Amazon. Você pode usar GameLift Anywhere a Amazon para integrar seus próprios recursos computacionais com a computação EC2 GameLift gerenciada pela Amazon para distribuir seus servidores de jogos em vários tipos de computação. Você também pode usar GameLift Anywhere a Amazon para testar iterativamente seus servidores de jogos sem fazer o upload da compilação GameLift para a Amazon a cada iteração.

Destaques:

- Novos tipos de GameLift Anywhere frota e computação da Amazon
- Registro de recursos GameLift Anywhere computacionais da Amazon
- Ciclo de iteração de teste aprimorado

O Amazon GameLift Server SDK 5.0.0 introduz melhorias no SDK do servidor existente e em um novo tipo de recurso, a computação. O Server SDK 5.0.0 oferece suporte à Amazon GameLift Anywhere e ao uso de seus próprios recursos computacionais para hospedagem de servidores de jogos.

Saiba mais:

- [Referência do SDK do servidor do Amazon GameLift](#)
- [Localização da frota](#)
- [Escolhendo os GameLift recursos computacionais da Amazon](#)
- [Crie uma GameLift Anywhere frota da Amazon](#)

25 de agosto de 2022: Amazon GameLift lança suporte para Locais Zones

Versões atualizadas do SDK: AWS SDK 1.9.333

GameLift A Amazon agora está disponível em oito Zonas Locais nos Estados Unidos, para que você possa posicionar suas frotas mais perto dos jogadores. Você pode usar todos os GameLift recursos gerenciados da Amazon com Zonas Locais adicionando as Zonas Locais às suas frotas.

As Zonas AWS Locais estendem recursos e serviços até a borda da nuvem, perto de grandes centros populacionais, industriais e de tecnologia da informação (TI). Isso significa que é possível implantar aplicativos que exigem latência de um dígito em milissegundos mais perto dos usuários finais ou dos data centers on-premises.

Saiba mais:

- [Zonas Locais](#)
- [Localização da frota](#)
- [Criar uma frota gerenciada pelo Amazon GameLift](#)

28 de junho de 2022: Amazon GameLift lança uma nova experiência de console opcional

O novo GameLift console da Amazon inclui essas melhorias:

- Navegação aprimorada — O novo painel de navegação facilita a navegação entre os recursos da Amazon GameLift .
- Página GameLift inicial da Amazon — A nova página inicial fornece links para documentação útil, exibe uma visão geral de alto nível da Amazon GameLift e fornece suporte por meio de links para documentação, perguntas frequentes e. AWS re:Post
- CloudWatch Métricas aprimoradas da Amazon — As GameLift métricas da Amazon agora estão disponíveis no GameLift console da Amazon e em seus CloudWatch painéis. Essa atualização também inclui novas métricas de desempenho, utilização e sessões de jogadores.

Saiba mais:

- [Visualizar os dados do jogo no console](#)
- [Gerenciar recursos de hospedagem do Amazon GameLift](#)
- [Construindo um FlexMatch casamenteiro](#)

15 de fevereiro de 2022: FlexMatch adiciona regra composta e melhorias adicionais

FlexMatch os usuários agora têm acesso aos seguintes recursos:

- Regra composta: foi adicionado suporte para regras compostas de criação de jogos para jogos de 40 ou menos jogadores. Agora é possível usar declarações lógicas para criar uma regra composta

para formar um jogo. Sem uma regra composta em seu conjunto de regras, para formar um jogo, todas as regras do conjunto de regras devem ser verdadeiras. Com regras compostas, você pode escolher quais regras aplicar usando os seguintes operadores lógicos: `and`, `or`, `not` e `xor`.

- Seleção flexível de equipes: expressões de propriedades de criação de jogos atualizadas para apoiar a seleção de um subconjunto de todas as equipes disponíveis.
- Listas de sequências mais longas: Aumentamos o número máximo de sequências de caracteres de 10 para 100 em uma lista de sequências de valores de atributos do jogador.

Saiba mais:

- [Guia do GameLift FlexMatch desenvolvedor da Amazon:](#)
 - [FlexMatch tipos de regras](#)
 - [FlexMatch expressões de propriedade](#)
- [AttributeValue: SL](#)

28 de outubro de 2021: Amazon GameLift adiciona suporte para frotas multirregionais na região Ásia-Pacífico (Osaka); Amazon FleetIQ adiciona suporte para processadores GameLift Graviton2 AWS

Versões atualizadas do SDK: AWS [SDK 1.9.133](#)

A Amazon agora GameLift está disponível na região Ásia-Pacífico (Osaka). Agora, os desenvolvedores de jogos podem implantar instâncias em Osaka usando uma frota GameLift multirregional.

Agora é possível usar servidores de jogos hospedados no Graviton2, com base na arquitetura de processador baseado em ARM, para obter maior desempenho a um custo menor quando comparado às opções de computação equivalentes baseadas em Intel.

Destaques:

- A Amazon agora GameLift está disponível na região Ásia-Pacífico (Osaka).
- Os grupos de servidores de jogos Amazon GameLift FleetIQ agora podem ser configurados para gerenciar as famílias de instâncias Graviton2 c6g, m6g e r6g.

Saiba mais:

- [Frota GameLift multirregional da Amazon](#)
- [CreateGameServerGroup](#)
- [AWS processador de grávitons](#)

20 de setembro de 2021: Amazon GameLift lança plug-in para Unity

O GameLift plug-in Amazon para Unity versão 1.0.0 contém bibliotecas e interface de usuário nativa que facilitam o acesso aos GameLift recursos da Amazon e a GameLift integração da Amazon ao seu jogo Unity. Você pode usar o GameLift plug-in Amazon para Unity para acessar as GameLift APIs da Amazon e implantar AWS CloudFormation modelos para cenários de jogos comuns. O plug-in também inclui um jogo de amostra que funciona com os cenários de amostra. Você pode usar o Amazon GameLift Local para ver as mensagens passadas entre o cliente do jogo e o servidor do jogo para saber como um jogo típico interage com a Amazon GameLift.

O plug-in para Unity é compatível com o Unity 2019.4 LTS e 2020.3 LTS.

Destaques:

- Crie, execute e modifique um jogo de amostra com cenários diferentes ou crie o seu próprio.
- Implante exemplos de AWS CloudFormation cenários para cenários de jogo típicos, incluindo somente autenticação, frota de região única, frotas multirregionais com fila e matchmaker personalizado, Spot Fleets com fila e matchmaker personalizado, e. FlexMatch

Saiba mais:

- [Integrando jogos com o GameLift plugin Amazon para Unity](#)

30 de junho de 2021: FlexMatch adiciona a regra BatchDistance

É possível usar o tipo de regra BatchDistance para especificar uma sequência de caracteres ou um atributo numérico, trazendo uma série de benefícios para cada segmento.

Destaques:

- Para jogos grandes (>40 jogadores), em vez de equilibrar uniformemente os jogadores apenas por habilidade, agora é possível obter o mesmo equilíbrio com base na habilidade, nos modos e nos mapas. Verifique se todos no jogo estejam em uma faixa de habilidade, agrupe vários atributos

numéricos, como liga ou estilo de jogo, e agrupe de acordo com atributos de sequência, como mapa ou modo de jogo. Também é possível criar expansões ao longo do tempo. Por exemplo, você pode criar uma expansão para permitir que uma faixa maior de nível de habilidade entre no jogo quanto mais tempo o jogador estiver esperando.

Para jogos com menos de 40 jogadores, é possível usar uma nova expressão de regras simplificada.

3 de junho de 2021: atualizações do SDK do cliente GameLift em tempo real da Amazon e do SDK do servidor

Versões atualizadas do SDK: SDK 1.2.0 de cliente em tempo real, SDK 3.4.0 de servidor para Unreal

Com essa atualização mais recente do SDK, agora é possível integrar o IL2CPP em seus aplicativos móveis que usam o SDK de cliente RTS e seguir as melhores práticas com estruturas. Agora você também pode criar o Amazon GameLift Server SDK para Unreal versão 4.26. Essa atualização contém componentes que se integram ao seu servidor de jogos Windows ou Linux, incluindo versões C++ e C# do Amazon GameLift Server SDK, Amazon GameLift Local e um plug-in Unreal Engine.

Destaques:

- Foi adicionado suporte para IL2CPP no SDK de cliente RTS e para criar bibliotecas nativas como estruturas, para que você possa criar clientes RTS para os dispositivos móveis mais recentes.
- É possível usar [DescribePlayerSessions\(\)](#) para obter informações de uma única sessão de jogador, para todas as sessões de jogador em uma sessão de jogo, ou para todas as sessões de jogador associadas a um único ID de jogador.
- É possível usar [GetInstanceCertificate\(\)](#) para recuperar o local do arquivo de um certificado TLS codificado em PEM associado à frota e às suas instâncias.
- Foi criado o suporte do SDK de servidor para Unreal versão 4.26.
- O SDK C# existente, versão 4.0.2, foi verificado como compatível com o Unity 2020.3. Nenhuma atualização do SDK é necessária.

Saiba mais:

- [Guia do GameLift desenvolvedor da Amazon:](#)
 - [DescribePlayerSessions\(\)](#)
 - [GetInstanceCertificate\(\)](#)

23 de março de 2021: Amazon GameLift adiciona notificações ao posicionamento da sessão de jogo

Versões atualizadas do SDK: AWS [SDK 1.8.168](#)

Agora é possível usar eventos para monitorar a atividade de posicionamento da sessão de jogo em uma fila de sessões de jogo. Crie um tópico do Amazon Simple Notification Service (Amazon SNS) para publicar notificações de eventos ou configure o rastreamento de eventos usando Eventos. CloudWatch

Destaques:

- Para cada fila, é possível definir uma sequência de texto personalizada para ser incluída em todas as mensagens do evento.
- Ao usar um tópico do Amazon SNS, é possível definir condições de acesso adicionais que limitam a publicação em filas específicas.

Saiba mais:

- Guia do GameLift desenvolvedor da Amazon:
 - [Configurar notificações de eventos para o posicionamento de sessões de jogo](#) (novo)
 - [Eventos de posicionamento de sessões de jogo](#) (novo)
- [Referência da API \(SDK da AWS\)](#)
 - Novos parâmetros da fila de sessões de jogo `NotificationTarget` e `CustomEventData: GameSessionQueue`, [CreateGameSessionQueueUpdateGameSessionQueue](#)
- [GameLiftFórum da Amazon](#)

16 de março de 2021: Amazon GameLift adiciona frotas multirregionais, seis novas regiões

Versões atualizadas do SDK: AWS [SDK 1.8.163](#)

A hospedagem GameLift gerenciada da Amazon agora está disponível em 21 AWS regiões. As novas regiões são Cidade do Cabo (af-south-1), Bahrein (me-south-1), Hong Kong (ap-east-1), Milão (eu-south-1), Paris (eu-west-3) e Estocolmo (eu-north-1).

Com o novo recurso de frotas em GameLift várias localizações da Amazon, agora você pode configurar uma única frota para hospedar seus servidores de jogos em qualquer uma ou em todas as 20 regiões GameLift suportadas pela Amazon (exceto a região de Pequim). Esse recurso visa reduzir significativamente o trabalho necessário para configurar e manter os recursos de GameLift hospedagem da Amazon globalmente. É possível criar frotas com vários locais nas seguintes AWS regiões: us-east-1 (Norte da Virgínia), us-west-2 (Oregon), eu-central-1 (Frankfurt), eu-west-1 (Irlanda), ap-southeast-2 (Sydney), ap-northeast-1 (Tóquio) e ap-northeast-2 (Seul). Em todas as outras regiões, é possível continuar configurando frotas de localização única conforme necessário. Todas as frotas criadas antes dessa versão são frotas de localização única. O uso de frotas com vários locais não afeta seus custos de hospedagem. GameLiftOs preços da Amazon são baseados no tipo, localização e volume de instâncias que você usa. (Para obter mais informações, consulte os [GameLiftpreços da Amazon](#).) AWS CloudFormation o suporte para frotas em vários locais estará disponível em breve.

Note

As frotas com vários locais não estão disponíveis nas regiões da China. GameLiftOs recursos da Amazon que residem nas regiões da China não podem interagir ou ser usados por recursos em outras GameLift regiões da Amazon.

Destaques:

- Com uma frota com vários locais, adicione explicitamente uma lista de locais remotos. A Amazon GameLift implanta instâncias do mesmo tipo e configuração, incluindo a configuração de construção e tempo de execução, na região de origem da frota e em todos os locais adicionados.
- Ajuste as configurações de capacidade e o dimensionamento para cada local de forma independente. As políticas de ajuste de escala automático se aplicam a uma frota inteira, mas é possível ativá-las ou desativá-las por local.
- Inicie novas sessões de jogo em locais específicos da frota. Ao usar filas de sessões de jogo ou criação de jogos para organizar sessões de jogo, agora é possível priorizar onde as novas sessões de jogo começam por local, custo de hospedagem e latência do jogador.
- Obtenha métricas de hospedagem no GameLift console da Amazon, agregadas para todos os locais em uma frota ou divididas por localização de cada frota.

Saiba mais:

- [Blog de tecnologia de jogos da Amazon](#)
- [Referência da API \(SDK da AWS \)](#)
 - Novas operações de localização da frota: [CreateFleetLocations](#), [DescribeFleetLocationAttributes](#), [DescribeFleetLocationCapacity](#), [DescribeFleetLocationUtilization](#), [DeleteFleetLocations](#)
 - Operações de frota atualizadas, com novo suporte a vários locais: [CreateFleet](#), [DescribeEC2UpdateFleetCapacityInstanceLimits](#), [DescribeInstancesStopFleetActionsStartFleetActions](#)
 - Operações de posicionamento de sessões de jogo atualizadas, com nova prioridade e capacidade de filtragem: [CreateGameSessionQueue](#), [DescribeGameSessionQueuesUpdateGameSessionQueue](#)
 - Operações de criação de sessão de jogo atualizadas, com novo suporte de localização: [CreateGameSession](#), [DescribeGameSessions](#), [DescribeGameSessionDetails](#), [SearchGameSessions](#)
- [Guia do GameLift desenvolvedor da Amazon:](#)
 - [Locais GameLift de hospedagem da Amazon](#) (atualizado)
 - [Guia de criação de frota do Amazon GameLift](#) (novo)
 - [Escalar a capacidade de hospedagem do Amazon GameLift](#) (atualizado)
 - [Criar uma fila de sessões de jogo](#) (novo)
 - [Visualizar detalhes da frota](#) (atualizado)
- [GameLiftFórum da Amazon](#)

9 de fevereiro de 2021: Amazon GameLift amplia o suporte para instâncias AMD, autônomas FlexMatch

Versões atualizadas do SDK: AWS [SDK 1.8.139](#)

Esta versão inclui as seguintes alterações:

- Os grupos de servidores de jogos Amazon GameLift FleetiQ agora podem ser configurados para gerenciar as famílias de instâncias AMD C5a, M5a e R5a. Os tipos de instância do Amazon EC2 compatíveis, conforme listados para o GameServerGroup [InstanceDefinition](#), agora incluem o seguinte:
 - c5a.large, c5a.xlarge, c5a.2xlarge, c5a.4xlarge, c5a.8xlarge, c5a.12xlarge, c5a.16xlarge, c5a.24xlarge

- m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
- r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge

Observação: no momento, as instâncias AMD para FleetiQ não estão disponíveis para uso na região da China (Pequim). AWS Para obter as informações mais recentes, consulte [Disponibilidade de atributos e diferenças de implementação](#).

- A hospedagem GameLift gerenciada de jogos da Amazon agora oferece suporte a instâncias da AMD na região da China (Pequim), operadas pela Sinnet. As novas famílias de instâncias AMD incluem M5a e R5a. Os tipos de instância EC2 compatíveis, conforme listados para frota [InstanceType](#), agora incluem o seguinte:
 - m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
 - r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge
- A Amazon agora GameLift FlexMatch pode ser usada como uma solução autônoma de matchmaking na região da China (Pequim), operada pela Sinnet. Os clientes podem criar um FlexMatch matchmaker na região de Pequim e configurar o [FlexMatchMode](#) parâmetro como STANDALONE. Para obter mais informações sobre FlexMatch, seja com a hospedagem GameLift gerenciada da Amazon ou com uma solução de GameLift hospedagem que não seja da Amazon, consulte o [Amazon GameLift FlexMatch Developer Guide](#).
- Ao configurar notificações de eventos para a Amazon GameLift FlexMatch, agora você pode designar um tópico FIFO do Amazon SNS como alvo da notificação. Para obter mais informações, consulte:
 - [MatchmakingConfiguration NotificationTarget](#), Referência de GameLift API da Amazon
 - [Configurar notificação de FlexMatch eventos](#), Amazon GameLift FlexMatch Developer Guide
 - [Apresentando o Amazon SNS FIFO — mensagens F irst-in-first-out Pub/Sub](#), blog de notícias AWS

22 de dezembro de 2020: o SDK GameLift do servidor Amazon é compatível com Unreal Engine 4.25 e Unity 2020

Versões atualizadas do SDK: Amazon GameLift Server SDK 4.0.2, versão 3.3.3 do plugin Unreal

A versão mais recente do Amazon GameLift Server SDK contém os seguintes componentes:

- O plug-in Unreal atualizado foi atualizado para compatibilidade com o Unreal Engine 4.25. A API não foi alterada.
- O SDK C# existente, versão 4.0.2, foi verificado como compatível com o Unity 2020. Nenhuma atualização do SDK é necessária.

Baixe a versão mais recente do Amazon GameLift Server SDK em [Amazon GameLift Getting Started](#).

24 de novembro de 2020: Amazon GameLift FlexMatch agora disponível para jogos hospedados em qualquer lugar

Versões atualizadas do SDK: AWS [SDK 1.8.95](#)

Amazon GameLift FlexMatch é um serviço personalizável de matchmaking para jogos multijogador. Projetado inicialmente para usuários da hospedagem GameLift gerenciada da Amazon, agora FlexMatch pode ser integrado a jogos que usam outros sistemas de hospedagem peer-to-peer, incluindo computação local proprietária e tipos primitivos de computação em nuvem. Jogos que usam o Amazon GameLift FleetIQ para hospedagem de jogos no Amazon EC2 agora podem implementar matchmaking com FlexMatch

FlexMatch fornece um algoritmo robusto de matchmaking e uma linguagem de regras que oferece ampla latitude para personalizar o processo de matchmaking para que os jogadores sejam combinados com base nas principais características do jogador e na latência relatada. Além disso, FlexMatch oferece um fluxo de trabalho de solicitação de matchmaking que suporta recursos como grupos de jogadores, aceitação de jogadores e preenchimento de partidas. Quando você usa FlexMatch com hospedagem GameLift gerenciada da Amazon ou servidores em tempo real, o matchmaker usa automaticamente GameLift a Amazon para encontrar recursos de hospedagem e iniciar uma nova sessão de jogo para partidas recém-formadas. Ao usar FlexMatch como um serviço independente, o matchmaker devolve os resultados da partida ao seu jogo, que pode então iniciar uma nova sessão de jogo usando sua solução de hospedagem.

As operações de API para FlexMatch fazem parte da API de GameLift serviços da Amazon, que está incluída no AWS SDK e no AWS Command Line Interface (AWS CLI). Essa versão inclui essas atualizações para oferecer suporte à criação de jogos independente:

- O recurso `MatchmakingConfiguration` da API tem as seguintes alterações:

- Nova propriedade, `FlexMatchMode` indica se o matchmaker está sendo usado com hospedagem GameLift gerenciada pela Amazon ou como matchmaking independente.
- A propriedade `GameSessionQueueArns` não é necessária quando o `FlexMatchMode` é definido como independente.
- Essas propriedades não são usadas com criação de jogos independente:
`AdditionalPlayerCount`, `BackfillMode`, `GameProperties`, `GameSessionData`.
- O atributo de preenchimento automático não está disponível com criação de jogos independente.

24 de novembro de 2020: instâncias AMD agora disponíveis na Amazon GameLift

Versões atualizadas do SDK: AWS [SDK 1.8.95](#)

A lista de tipos de instância do Amazon EC2 suportados pela Amazon GameLift agora inclui três novas famílias de instâncias: C5a, M5a e R5a. Essas famílias consistem em instâncias AMD otimizadas para computação que são alimentadas por processadores AMD EPYC executados em frequências de até 3,3 GHz. As instâncias da AMD são compatíveis com x86; os jogos atualmente em execução na Amazon GameLift podem ser implantados nos tipos de instância da AMD sem alteração. As novas instâncias estão disponíveis nas seguintes AWS regiões: Leste dos EUA (Norte da Virgínia e Ohio), Oeste dos EUA (Oregon e Norte da Califórnia), Canadá Central (Montreal), América do Sul (São Paulo), UE Central (Frankfurt), Oeste da UE (Londres e Irlanda), Ásia-Pacífico Sul (Mumbai), Nordeste da Ásia-Pacífico (Seul e Tóquio) e Sudeste da Ásia-Pacífico (Cingapura e Sydney).

As novas instâncias AMD incluem:

- c5a.large, c5a.xlarge, c5a.2xlarge, c5a.4xlarge, c5a.8xlarge, c5a.12xlarge, c5a.16xlarge, c5a.24xlarge
- m5a.large, m5a.xlarge, m5a.2xlarge, m5a.4xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
- r5a.large, r5a.xlarge, r5a.2xlarge, r5a.4xlarge, r5a.8xlarge, r5a.12xlarge, r5a.16xlarge, r5a.24xlarge

Saiba mais:

- [Blog de tecnologia de jogos da Amazon](#)
- [Preços de GameLift instâncias da Amazon](#)
- [Instâncias do Amazon EC2 com processadores AMD EPYC](#)

- [GameLiftFórum da Amazon](#)

11 de novembro de 2020: atualização da versão do SDK GameLift do servidor Amazon

Versões atualizadas do SDK: Amazon GameLift Server SDK 4.0.2

A nova versão 4.0.2 do SDK de servidor corrige um problema conhecido com a operação `StartMatchBackfill()` da API. Essa operação agora retorna uma resposta correta para uma solicitação de preenchimento de jogos.

O problema não afetou o processo de preenchimento de jogos e não há alteração na forma como esse recurso funciona. O problema pode ter afetado as mensagens de log e o tratamento de erros nas solicitações de preenchimento de jogos.

Baixe a versão mais recente do Amazon GameLift Server SDK em [Amazon GameLift Getting Started](#).

5 de novembro de 2020: novas personalizações de FlexMatch algoritmos

FlexMatch os usuários agora podem ajustar os seguintes comportamentos padrão para o processo de matchmaking. Essas personalizações são definidas em um conjunto de regras de criação de jogos. Não há alterações nos GameLift SDKs da Amazon.

- **Priorize os tickets de preenchimento:** é possível escolher entre aumentar ou diminuir a forma como os tickets de preenchimento de jogos são priorizados ao pesquisar jogos aceitáveis. Priorizar os tickets de preenchimento é útil quando o recurso de preenchimento automático está ativado. Use a propriedade `backfillPriority` do algoritmo.
- **Pré-classifique para otimizar a consistência e a eficiência do jogo:** configure seu matchmaker para pré-classificar o pool de tickets antes de agrupar os tickets para avaliação. Ao pré-classificar os tickets com base nos principais atributos dos jogadores, os jogos resultantes tendem a ter jogadores mais semelhantes nesses atributos. Também é possível aumentar a eficiência no processo de avaliação pré-classificando os mesmos atributos usados nas regras de jogo. Use a propriedade `sortByAttributes` do algoritmo com a propriedade `strategy` definida como “classificada”.
- **Ajuste como os tempos de espera da expansão são acionados:** escolha entre ativar expansões com base na idade do ticket mais novo (padrão) ou do mais antigo em um jogo incompleto. O acionamento do ticket mais antigo tende a concluir as jogos mais rapidamente, enquanto

o acionamento do ticket mais novo leva a uma maior qualidade da jogo. Use a propriedade `expansionAgeSelection` do algoritmo.

17 de setembro de 2020: Amazon GameLift atualiza o SDK do servidor

Versões atualizadas do SDK: Amazon GameLift Server SDK 4.0.1

O novo SDK de servidor contém as seguintes atualizações:

- Versão 4.0.1 da API C#
 - A operação [TerminateGameSession\(\)](#) da API não é mais compatível. Substitua por uma chamada para [ProcessEnding\(\)](#) para encerrar a sessão do jogo e o processo do servidor.
 - Um problema conhecido com a operação [GetInstanceCertificate\(\)](#) foi corrigido.
 - A operação [GetTerminationTime\(\)](#) agora retorna um valor do tipo de dados `AwsDateTimeOutcome`.
- Versão 3.4.1 da API C#
 - A operação [TerminateGameSession\(\)](#) não é mais compatível. Substitua por uma chamada para [ProcessEnding\(\)](#) para encerrar a sessão do jogo e o processo do servidor.
- Plug-in do Unreal Engine versão 3.3.2
 - A operação [TerminateGameSession\(\)](#) não é mais compatível. Substitua por uma chamada para [ProcessEnding\(\)](#) para encerrar a sessão do jogo e o processo do servidor.
 - A operação `OnUpdateGameSession` de retorno de chamada é adicionada a [FProcessParameters](#) para oferecer suporte ao preenchimento de jogos.

Baixe a versão mais recente do Amazon GameLift Server SDK em [Amazon GameLift Getting Started](#).

27 de agosto de 2020: Amazon GameLift FleetIQ para hospedagem de jogos com o Amazon EC2 (disponibilidade geral)

Versões atualizadas do SDK: AWS [SDK 1.8.36](#)

A solução Amazon GameLift FleetIQ para hospedagem de jogos de baixo custo e baseada em nuvem no Amazon EC2 já está disponível ao público em geral. O Amazon GameLift FleetIQ oferece aos desenvolvedores a capacidade de hospedar servidores de jogos diretamente nas instâncias spot do Amazon EC2, otimizando sua viabilidade para hospedagem de jogos. Os desenvolvedores de

jogos podem usar o Amazon GameLift FleetIQ com novos jogos ou para complementar a capacidade dos jogos existentes. Essa solução suporta o uso de contêineres ou outros AWS serviços, como AWS Shield e Amazon Elastic Container Service (Amazon ECS).

Esta versão de disponibilidade geral inclui as seguintes atualizações da solução Amazon GameLift FleetIQ:

- A nova operação de API `DescribeGameServerInstances` retorna informações, incluindo status, sobre todas as instâncias ativas de um grupo de servidores de GameLift jogos Amazon FleetIQ.
- A nova estratégia de balanceamento, `ON_DEMAND_ONLY`, configura um grupo de servidores de jogos para usar somente instâncias sob demanda. É possível atualizar a estratégia de balanceamento de um grupo de servidores de jogos a qualquer momento, possibilitando alternância entre o uso de instâncias spot e instâncias sob demanda, conforme necessário.
- Os seguintes elementos de pré-visualização foram descartados para disponibilidade geral:
 - Uso de chaves de classificação personalizadas para recursos do servidor de jogos. Os servidores de jogos podem ser classificados com base no carimbo de data e hora do registro.
 - Marcação de recursos do servidor de jogos.

16 de abril de 2020: Amazon GameLift atualiza o SDK do servidor para Unity e Unreal Engine

Versões atualizadas do SDK: Amazon GameLift Server SDK 4.0.0, Amazon Local 1.0.5 GameLift

A versão mais recente do Amazon GameLift Server SDK contém os seguintes componentes atualizados:

- SDK C# versão 4.0.0 atualizado para Unity 2019
- Plug-in do Unreal versão 3.3.1 atualizado para Unreal Engine versões 4.22, 4.23 e 4.24.
- A versão 1.0.5 do Amazon GameLift Local foi atualizada para testar integrações que usam o SDK do servidor C# versão 4.0.0.

Baixe a versão mais recente do Amazon GameLift Server SDK em [Amazon GameLift Getting Started](#).

2 de abril de 2020: Amazon GameLift FleetIQ disponível para hospedagem de jogos no EC2 (versão prévia pública)

Versões atualizadas do SDK: AWS [SDK 1.7.310](#)

O recurso Amazon GameLift FleetIQ otimiza a viabilidade de instâncias spot de baixo custo para uso com hospedagem de jogos. Esse recurso agora é estendido para clientes que desejam gerenciar seus recursos de hospedagem diretamente, em vez de por meio do GameLift serviço gerenciado da Amazon. Essa solução suporta o uso de contêineres ou outros AWS serviços, como AWS Shield e Amazon Elastic Container Service (Amazon ECS).

Saiba mais:

GameTech publicação de [blog](#) no Amazon GameLift FleetIQ

19 de dezembro de 2019: gerenciamento aprimorado de AWS recursos para GameLift recursos da Amazon

Versões atualizadas do SDK: AWS [SDK 1.7.249](#)

Agora você pode aproveitar as ferramentas de gerenciamento de AWS recursos com GameLift os recursos da Amazon. Em particular, todos os principais GameLift recursos da Amazon — builds, scripts, frotas, filas de sessões de jogo, configurações de matchmaking e conjuntos de regras de matchmaking — agora recebem valores de Amazon Resource Name (ARN). O ARN de um recurso fornece um identificador consistente que é exclusivo em todas as AWS regiões. Eles podem ser usados para criar políticas de permissões específicas de recursos AWS Identity and Access Management (IAM). Agora os recursos recebem um ARN e o identificador de recurso pré-existente, que não é específico para a região.

Além disso, GameLift os recursos da Amazon agora oferecem suporte à marcação. Você pode usar tags para organizar recursos, criar políticas de permissões do IAM para gerenciar o acesso a grupos de recursos, personalizar detalhamentos de AWS custos etc. Ao gerenciar tags para GameLift recursos da Amazon, use as ações de GameLift API da Amazon `TagResource()`, `UntagResource()`, `ListTagsForResource()` e.

Saiba mais:

- [TagResource](#) na Amazon GameLift API Reference
- [Marcar recursos da AWS](#) na Referência geral da AWS

- [Nomes de recursos da Amazon](#) na Referência geral da AWS

14 de novembro de 2019: Novos modelos do AWS CloudFormation , atualizações na região da China (Pequim)

Versões atualizadas do SDK: AWS [SDK 1.7.210](#)

AWS CloudFormation modelos para Amazon GameLift

Agora, GameLift os recursos da Amazon podem ser criados e gerenciados por meio do AWS CloudFormation. Os modelos existentes de AWS CloudFormation construção e frota foram atualizados para se alinharem aos recursos atuais, e novos modelos agora estão disponíveis para scripts, filas, configurações de matchmaking e conjuntos de regras de matchmaking. AWS CloudFormation os modelos simplificam muito a tarefa de gerenciar grupos de AWS recursos relacionados, principalmente ao implantar jogos em várias regiões.

Saiba mais:

- [Referência GameLift de tipo de recurso da Amazon](#) no Guia AWS CloudFormation do usuário
- [Gerenciar recursos usando o AWS CloudFormation](#) no Amazon GameLift Developer Guide

Glossário do AWS

Para obter a terminologia mais recente da AWS, consulte o [glossário da AWS](#) na Referência do Glossário da AWS.

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.