AWS Livro branco

Lente da indústria de jogos

Lente da indústria de jogos: AWS Livro branco

Table of Contents

Lente da indústria de jogos — AWS Well-Architected Framework

Data de publicação: 9 de dezembro de 2025 (Revisões do documento)

O AWS Well-Architected Framework ajuda os arquitetos de nuvem a criar uma infraestrutura segura, de alto desempenho, resiliente e eficiente para seus aplicativos e cargas de trabalho. Com base em seis pilares — excelência operacional, segurança, confiabilidade, eficiência de desempenho, otimização de custos e sustentabilidade — a Well-Architected fornece uma abordagem consistente para clientes AWS e parceiros avaliarem arquiteturas, corrigirem riscos e implementarem projetos que agreguem valor comercial.

Nessa lente, nos concentramos em como projetar, implantar e arquitetar suas cargas de trabalho de jogos no Nuvem AWS. Definimos componentes, exploramos cenários comuns de carga de trabalho e descrevemos princípios de design que ajudam você a aplicar o Well-Architected Framework. Recomendamos que você comece a projetar sua arquitetura considerando as melhores práticas e questões do whitepaper do AWS Well-Architected Framework. Este documento fornece as melhores práticas complementares para clientes do setor de jogos.

Essa lente especifica as melhores práticas destinadas a abordar as características exclusivas de criar e operar jogos na nuvem com base em nossa experiência de trabalho com desenvolvedores e editores da indústria de jogos em todo o mundo. Ele fornece orientação sobre como projetar e operar seu ambiente para que ele seja otimizado em termos de custos e escalável para flutuações na demanda global de players. Essa lente também fornece orientação para proteger sua infraestrutura de jogo e ajustar o desempenho para proporcionar uma experiência positiva ao jogador.

Este documento é destinado a pessoas em cargos de tecnologia, como diretores de tecnologia (CTOs), diretores técnicos de estúdios de jogos, arquitetos, desenvolvedores e membros da equipe de operações. Depois de ler este documento, você entenderá as AWS melhores práticas e estratégias a serem usadas ao projetar arquiteturas para jogos.

Disponibilidade de lentes

As lentes personalizadas ampliam a orientação de melhores práticas fornecida pela AWS Well-Architected Tool. AWS WA Tool permite que você crie suas próprias lentes personalizadas ou use lentes criadas por outras pessoas que foram compartilhadas com você.

Para começar a analisar a carga de trabalho de seus jogos, baixe e importe o Games Industry Lens do repositório público de lentes AWS Well-Architected Tool personalizadas da AWS Well-Architected. GitHub

Princípios de design

O AWS Well-Architected Framework identifica os seguintes princípios gerais de design para facilitar um bom design na nuvem para cargas de trabalho de jogos:

- Entenda o comportamento e os padrões de uso dos jogadores para evoluir o jogo e ajudar a protegê-los: Para promover melhorias contínuas em seu jogo e gerenciar com eficácia a experiência do jogador, é importante ter visibilidade de como os jogadores interagem com o jogo em si e com o resto dos jogadores. Isso ajuda você a entender como melhorar o jogo, gerenciar custos e monitorar e reagir a atividades de uso não autorizadas que representam riscos à experiência do jogador.
- Use tecnologias que simplifiquem as operações do jogo e aumentem a velocidade de desenvolvimento: priorize a adoção de tecnologias que possam melhorar a velocidade e reduzir a sobrecarga operacional de oferecer novos recursos e melhorias aos jogadores. Os jogos são baseados em acertos e há muitas opções que os jogadores devem considerar, portanto, agir rapidamente e se adaptar às mudanças é fundamental para o sucesso de um jogo. Considere se você se sente confortável em operar seu próprio software ou se prefere adotar um serviço gerenciado comparável da AWS AWS Partners ou de ambos.
- Otimize sua arquitetura para melhorar as métricas que refletem a experiência real do jogador: à medida que você adapta e evolui sua arquitetura ao longo do tempo, pense em como essas melhorias e mudanças afetarão a experiência do jogador. As cargas de trabalho de jogos devem ser capazes de suportar e minimizar o impacto das falhas para bloquear interrupções generalizadas na jogabilidade. Os recursos e sistemas do jogo que não dependem criticamente uns dos outros devem ser desacoplados para reduzir o impacto das falhas e isolar os problemas que afetam os jogadores.
- Projete a infraestrutura para atender ao pico de simultaneidade de jogadores e escale dinamicamente conforme necessário: a infraestrutura deve ser projetada para ser dimensionada para atender à demanda dos jogadores. Métricas, como simultaneidade de sessões de jogadores e número de logins, podem ser usadas para escalar preventivamente antes que os sistemas fiquem sobrecarregados. Métricas reativas de utilização do sistema, como consumo de CPU e memória, podem ser usadas para escalar depois que os sistemas ficarem sobrecarregados. Ao escalar dinamicamente sua infraestrutura, você pode reduzir os custos de operação do seu jogo.
- Implemente runbooks para melhorar as operações do jogo: Os runbooks operacionais devem ser usados para gerenciar de forma consistente as tarefas recorrentes de operações do jogo. Devem existir runbooks para fluxos de trabalho comuns de operações de jogos, como investigar

e responder aos relatórios dos jogadores, gerenciar atividades de pré-escalonamento da infraestrutura para se preparar para eventos previstos de grande escala, como lançamentos de novas temporadas e lançamentos de conteúdo de jogos, e para abordar atividades típicas de manutenção de jogos.

Cenários

Nesta seção, abordamos vários cenários que são comuns em uma arquitetura de jogo. Cada cenário inclui as características comuns que orientam o design e um exemplo de diagrama de arquitetura de referência.

Cenários

Hospedagem de jogos para jogabilidade síncrona em tempo real

A jogabilidade síncrona em tempo real permite que dois ou mais jogadores participem e interajam em um jogo simultaneamente, onde o estado da jogabilidade é compartilhado entre os jogadores conectados para criar a experiência mais próxima possível de uma experiência em tempo real. Exemplos de jogos síncronos incluem jogos de tiro em primeira pessoa, jogos online multijogador massivo (MMOG), jogos de esportes e ação ou um jogo online em que dois ou mais jogadores devem estar conectados para compartilhar a experiência de jogo quase em tempo real.

As características das arquiteturas de jogo síncrono em tempo real incluem:

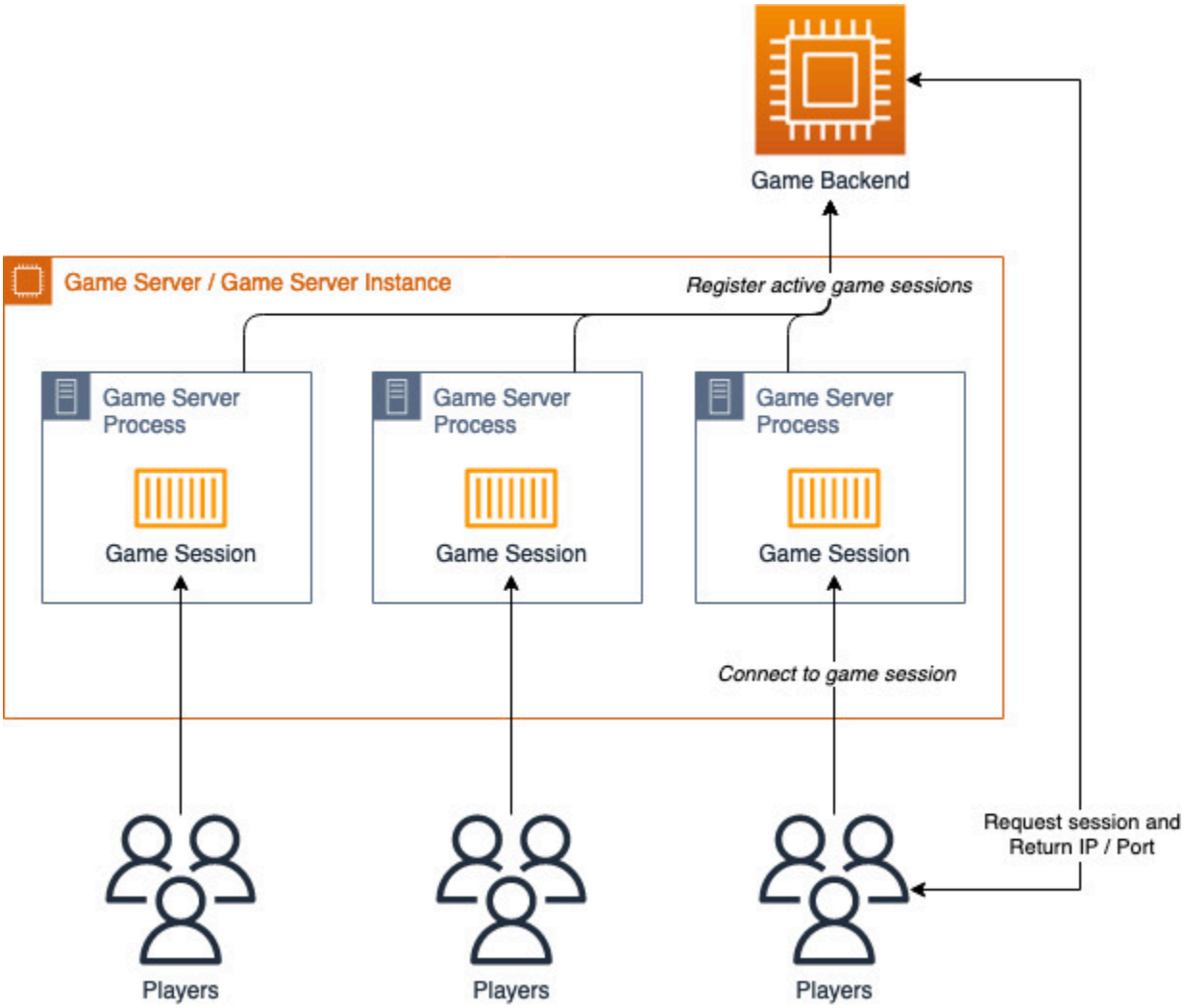
- Alguns jogos podem ser hospedados como sessões de jogo por meio de processos de servidor de jogos executados em servidores dedicados. Alguns jogos podem usar arquiteturas do tipo P2P que empregam utilitários de passagem de sessão mais leves para NAT (STUN) ou Traversal usando relés em servidores NAT (TURN). Independentemente do tipo de servidor envolvido, os servidores de jogos são hospedados em vários data centers e Regiões da AWS globalmente.
- Os clientes do jogo podem participar de uma sessão de jogo solicitando uma partida de um serviço centralizado de matchmaking hospedado no sistema de back-end do jogo ou selecionando uma partida em uma lista predefinida de servidores de jogos disponíveis. O cliente do jogo é fornecido com um endereço IP e uma porta à qual se conectar.
- Muitos jogos síncronos são sensíveis à latência, como jogos de tiro em primeira pessoa e jogos online multijogador massivos. Eles provavelmente incluirão algoritmos como retrocesso e dilatação

do tempo para minimizar os efeitos do atraso, mas também podem ter uma tolerância de latência predefinida que é cuidadosamente medida e otimizada para reduzir a experiência de atraso que às vezes pode ocorrer para jogadores em situações de alta latência. Essas informações de latência são determinadas pela instrumentação dos clientes do jogo para fazer ping no servidor de jogo disponível Regiões da AWS para capturar métricas como latência, instabilidade de rede e outras métricas importantes para a experiência de jogo. Essas métricas são enviadas para um serviço central de coleta de métricas no sistema de back-end do jogo para que as transmissões operacionais ao vivo possam monitorar a integridade do jogo. Durante o processo de matchmaking, os clientes do jogo fornecem seus dados de latência atuais como um dos parâmetros de solicitação ao solicitar uma partida, e o serviço de matchmaking pode usar esses dados de latência como uma das variáveis ao selecionar um servidor de jogo para hospedar o jogador.

- Normalmente, a jogabilidade é conduzida por meio de uma combinação de protocolos (como servidores de jogos usando mensagens mais rápidas baseadas em UDP com matchmaking, autenticação e outro tráfego cliente-servidor usando HTTPS).
- Os servidores de jogos empregam algoritmos e design para minimizar o tráfego cliente-servidor, como transformações de streaming, deltas e compressão de dados.
- Os servidores de jogos são alvos frequentes de atividades maliciosas e devem ser protegidos com uma solução de proteção DDo S, como AWS Shield Advanced.

Processos do servidor de jogos

O diagrama a seguir ilustra uma arquitetura típica de servidor de jogos. Ele descreve a relação lógica entre uma instância do servidor de jogos e os processos do servidor de jogos que hospedam as sessões de jogo.



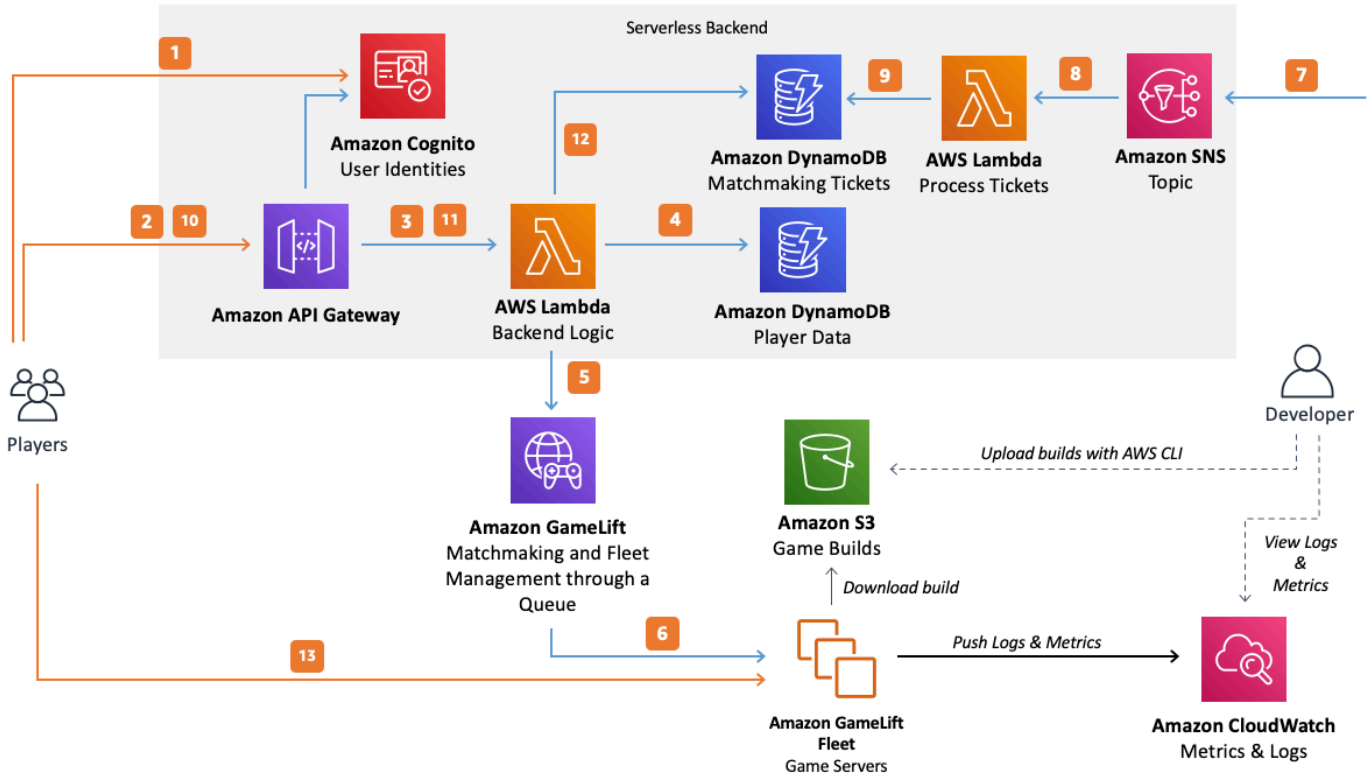
Arquitetura lógica de servidor de jogos

- Uma EC2 instância da Amazon é usada como servidor de jogos, também conhecida como instância de servidor de jogos. O servidor do jogo hospeda um ou mais processos do servidor de jogos, e cada um está executando uma cópia da versão do seu servidor de jogos. Normalmente, vários processos do servidor de jogos são executados em uma instância do servidor de jogos para utilizar os recursos computacionais de forma eficiente e reduzir custos. Quando uma sessão de jogo está ativa e pronta para hospedar sessões de jogadores, seu status é atualizado com o back-end do jogo (geralmente um serviço de matchmaking) para que possa começar a ser usada para hospedar jogadores.
- O back-end do jogo pode fornecer ao jogador o endereço IP e a porta do servidor que está hospedando uma sessão de jogo para que ele possa se conectar para jogar.

Hospedagem de servidor de jogos baseada em sessão com back-end sem servidor

Ao desenvolver uma arquitetura para seu jogo, considere os recursos e capacidades de que você precisa e o nível de sobrecarga de gerenciamento operacional que você está preparado para assumir. Para oferecer o melhor equilíbrio entre facilidade de operação e flexibilidade, você pode criar seu jogo usando serviços gerenciados de provedores de nuvem. Os serviços gerenciados oferecem a você o controle para desenvolver e personalizar seus próprios recursos de jogo personalizados, além de reduzir a carga de implantação e gerenciamento da infraestrutura.

Hospedar um jogo multijogador baseado em sessão requer uma infraestrutura de servidor para hospedar os processos do servidor do jogo, bem como um back-end escalável para matchmaking e gerenciamento de sessões. A arquitetura de referência a seguir mostra como a hospedagem GameLift gerenciada e um back-end sem servidor da Amazon podem ser usados para gerenciar seus jogos baseados em sessões.



Hospedagem GameLift gerenciada da Amazon para jogos baseados em sessões

O diagrama descreve o processo de fazer com que os jogadores entrem em jogos GameLift executados na hospedagem gerenciada de jogos. Inclui as seguintes etapas:

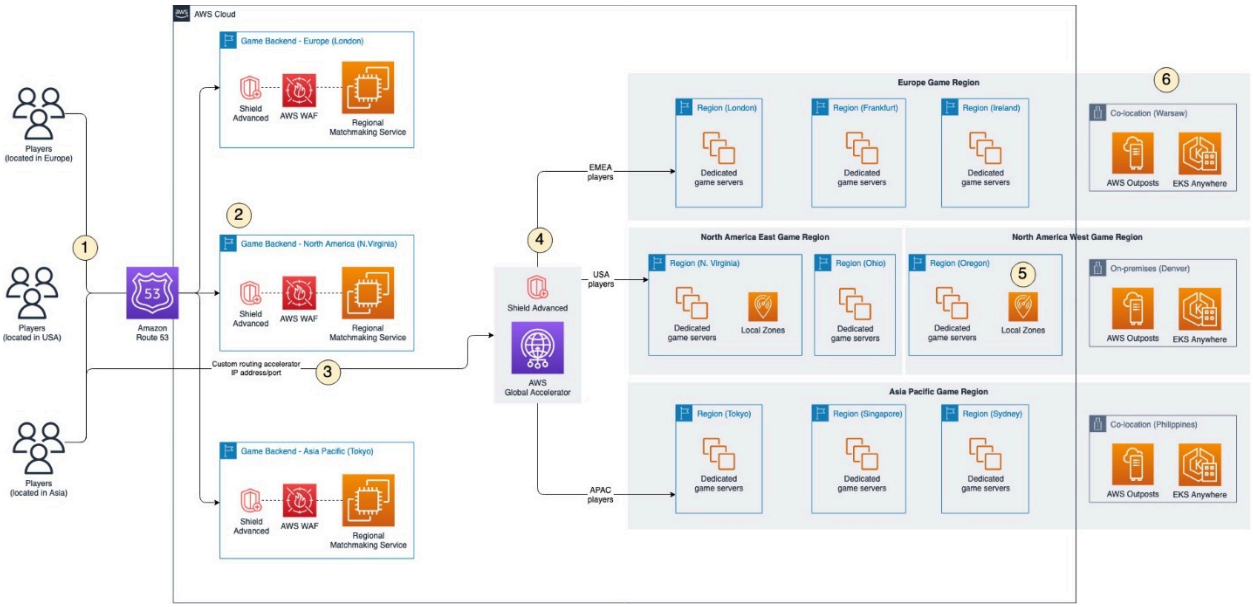
1. O cliente do jogo solicita uma identidade do Amazon Cognito de um pool de identidades do Amazon Cognito. Opcionalmente, isso pode ser conectado a provedores de identidade externos.
2. O cliente do jogo recebe credenciais de acesso temporárias e solicita uma sessão de jogo por meio de um Amazon API Gateway assinando a solicitação com as credenciais do Amazon Cognito.
3. O API Gateway invoca uma AWS Lambda função.
4. A função Lambda solicita dados do player de uma tabela do Amazon DynamoDB. A identidade do Amazon Cognito é usada para solicitar com segurança os dados corretos do jogador porque a identidade autenticada é fornecida nos dados de contexto da solicitação.
5. Usando os dados corretos do jogador para obter informações adicionais (como o nível de habilidade do jogador), a função Lambda solicita uma partida por meio GameLift FlexMatch de matchmaking. Você pode definir uma configuração de FlexMatch matchmaking com documentos de configuração baseados em JSON. O cliente do jogo pode gerar métricas de latência fazendo ping nos endpoints do servidor em várias regiões, e os dados de latência podem ser usados para apoiar o matchmaking baseado em latência.
6. Depois de FlexMatch combinar um grupo adequado de jogadores com latência adequada para uma região, ele solicita uma colocação na sessão de jogo por meio de uma GameLift fila. A fila contém frotas com uma ou mais localizações regionais registradas.
7. Quando a sessão é colocada em um dos locais da frota, uma notificação de evento é enviada para um tópico do Amazon SNS.
8. Uma função Lambda receberá o evento do Amazon SNS e o processará.
9. Se a mensagem do Amazon SNS for um MatchmakingSucceeded evento, a função Lambda grava o resultado no DynamoDB com a porta do servidor e o endereço IP. Um valor time-to-live (TTL) é usado para garantir que os tickets de matchmaking sejam excluídos do DynamoDB quando não forem mais necessários.
10. O cliente do jogo faz uma solicitação assinada ao API Gateway para verificar o status do tíquete de matchmaking em um intervalo específico.
11. O API Gateway invoca uma função Lambda que verifica o status do ticket de matchmaking.
12. A função Lambda verifica o DynamoDB para determinar se o ticket foi bem-sucedido. Se for bem-sucedida, a função Lambda envia o endereço IP, a porta e o ID da sessão do player de volta ao cliente. Se o ticket falhar, a função Lambda enviará uma resposta declarando que a partida 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 back-end. Ele envia o ID da sessão do jogador para o servidor do jogo, e o servidor do jogo o valida usando o Amazon GameLift Server SDK.

Como alternativa, você pode modificar a arquitetura anterior para usar o API Gateway WebSockets com a Amazon GameLift. Nessa abordagem, a comunicação entre o cliente do jogo e o serviço de back-end do jogo ocorre usando uma implementação WebSocket baseada. Essa implementação pode ser usada para que a função Lambda do backend do jogo inicie uma mensagem do lado do servidor para o cliente do jogo em vez de implementar WebSocket um modelo de pesquisa.

Arquitetura multirregional e híbrida para jogos de baixa latência

Esta seção descreve uma arquitetura multirregional e híbrida para jogos de baixa latência.



Reduzindo a latência com aceleração de rede e servidores de jogos implantados globalmente

1. Os jogadores de um jogo disponível globalmente podem se originar de qualquer lugar. Quando um jogador solicita uma sessão de jogo ou partida, seu cliente de jogo envia uma solicitação para o serviço de back-end do jogo registrado no Amazon Route 53. O roteamento baseado em latência do Route 53 pode ser usado para direcionar o jogador para o back-end de jogo mais próximo disponível.
2. O back-end do jogo é implantado em várias populações de jogadores Regiões da AWS mais próximas. Cada back-end do jogo inclui um serviço regional de matchmaking que encontra uma sessão de jogo de todas as regiões do jogo. Embora a solicitação de matchmaking de um

jogador seja processada por um serviço regional de matchmaking próximo a ele, o serviço de matchmaking é capaz de encaminhar um jogador para uma sessão de jogo em outra região do jogo, se necessário. Essa ação melhora a resiliência e o desempenho. Além disso, cada serviço de back-end do jogo usa AWS WAF e AWS Shield Advanced fornece filtragem da web de camada 7, controle de bots e proteções contra ataques de negação de serviço (S) de distribuição. DDo Você tem muitas opções para criar seu serviço de back-end de jogos, como sem servidor, contêineres, EC2 instâncias ou hospedagem do serviço de back-end de jogos em seus próprios data centers.

3. Para melhorar a experiência dos jogadores reduzindo a latência e a instabilidade da rede, um acelerador de roteamento personalizado é implantado usando o AWS Global Accelerator, que otimiza automaticamente o roteamento do tráfego do cliente do jogo para o servidor do jogo usando a rede global. AWS Você configura o acelerador de roteamento personalizado para mapear as portas do ouvinte do Global Accelerator para a porta de instância do seu servidor de jogos. EC2 O cliente do jogo se conecta ao IP e à porta do Global Accelerator que serve como proxy e direciona deterministicamente os jogadores para o IP e a porta corretos do servidor de jogo que está hospedando a sessão do jogo.
4. Seu jogo inclui regiões lógicas de jogo fáceis de usar que representam uma coleção de locais de hospedagem de servidores de jogos que estão geograficamente próximos uns dos outros, como América do Norte ou Ásia-Pacífico. Para reduzir a latência e aumentar a cobertura geográfica, uma combinação de diferentes soluções de hospedagem de servidores de jogos pode ser usada para melhorar a experiência do jogador. Priorize o uso de Regiões da AWS sempre que possível, pois esses locais têm todos os recursos e contêm a maior capacidade ocupada.
5. Use as Zonas AWS Locais para hospedar servidores de jogos em localizações geográficas de jogadores mal atendidos, onde você não tem instalações de hospedagem existentes ou não Região da AWS estão disponíveis.
6. Implemente AWS Outposts em seus datacenters locais e provedores de co-localização existentes para criar um plano de controle e uma experiência de gerenciamento ininterruptos em cada local de implantação usando racks totalmente gerenciados e servidores montados em rack. AWS Outposts também são benéficos se você não tiver capacidade de servidor existente em seu ambiente local. No entanto, se você quiser uma implementação híbrida com software executado em sua própria infraestrutura de servidor existente, você deve usar o Amazon EKS Anywhere, que permite criar e executar clusters Kubernetes em sua própria infraestrutura com conectividade com o Amazon EKS. Isso fornece uma visão consistente do console de seus clusters do Kubernetes no local.

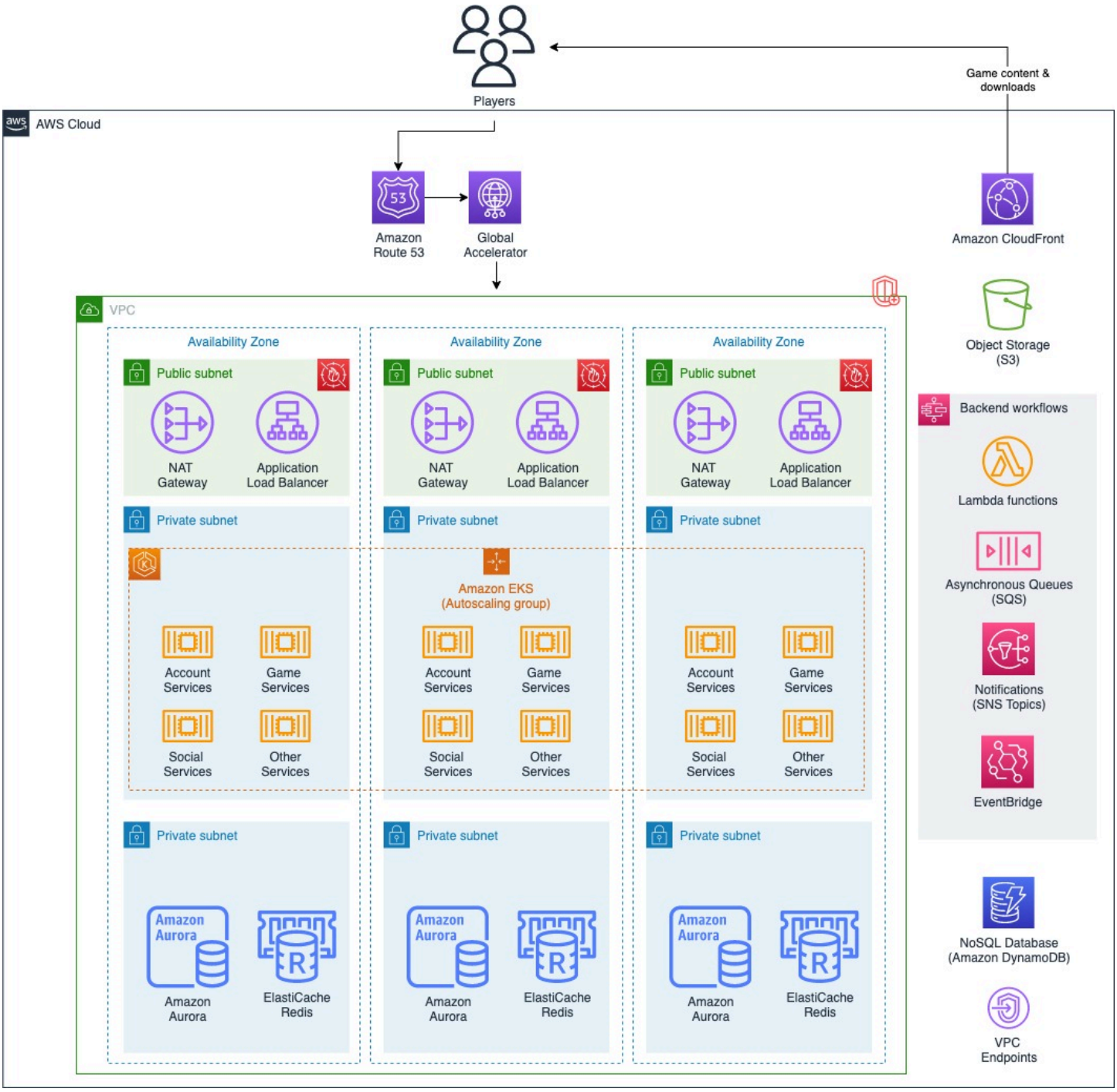
Backends de jogos

Os back-ends de jogos são usados para gerenciar o estado do jogo e do jogador, bem como integrar recursos sociais e de nível de sistema ao jogo que oferecem suporte à experiência de jogo. Gerenciamento de perfil de jogador, armazenamento de itens e inventário, estatísticas e tabelas de classificação são exemplos de serviços hospedados em back-ends de jogos.

Os back-ends de jogos geralmente são criados como REST APIs e são acessados por clientes usando HTTPS. No entanto, outras abordagens também são comuns, como a WebSockets que fornecem canais bidirecionais para casos de uso, como notificações de clientes para bate-papo e presença no jogo. Os back-ends de jogos podem ser implantados usando uma variedade de arquiteturas de implantação diferentes, incluindo o uso de instâncias, contêineres ou uma arquitetura sem servidor.

Arquitetura de back-end de jogos baseada em contêineres

Esta seção descreve uma arquitetura de back-end de jogos baseada em contêineres.



Hospedando um back-end de jogos usando contêineres

- Os jogadores acessam o jogo usando o software cliente do jogo, que pode ser distribuído por meio de um sistema de jogos, uma loja digital ou um download direto de uma rede de distribuição de conteúdo (CDN), como a Amazon. CloudFront CDNsforneça armazenamento em cache em locais periféricos para acelerar o desempenho dos usuários que baixam conteúdo. Por exemplo,

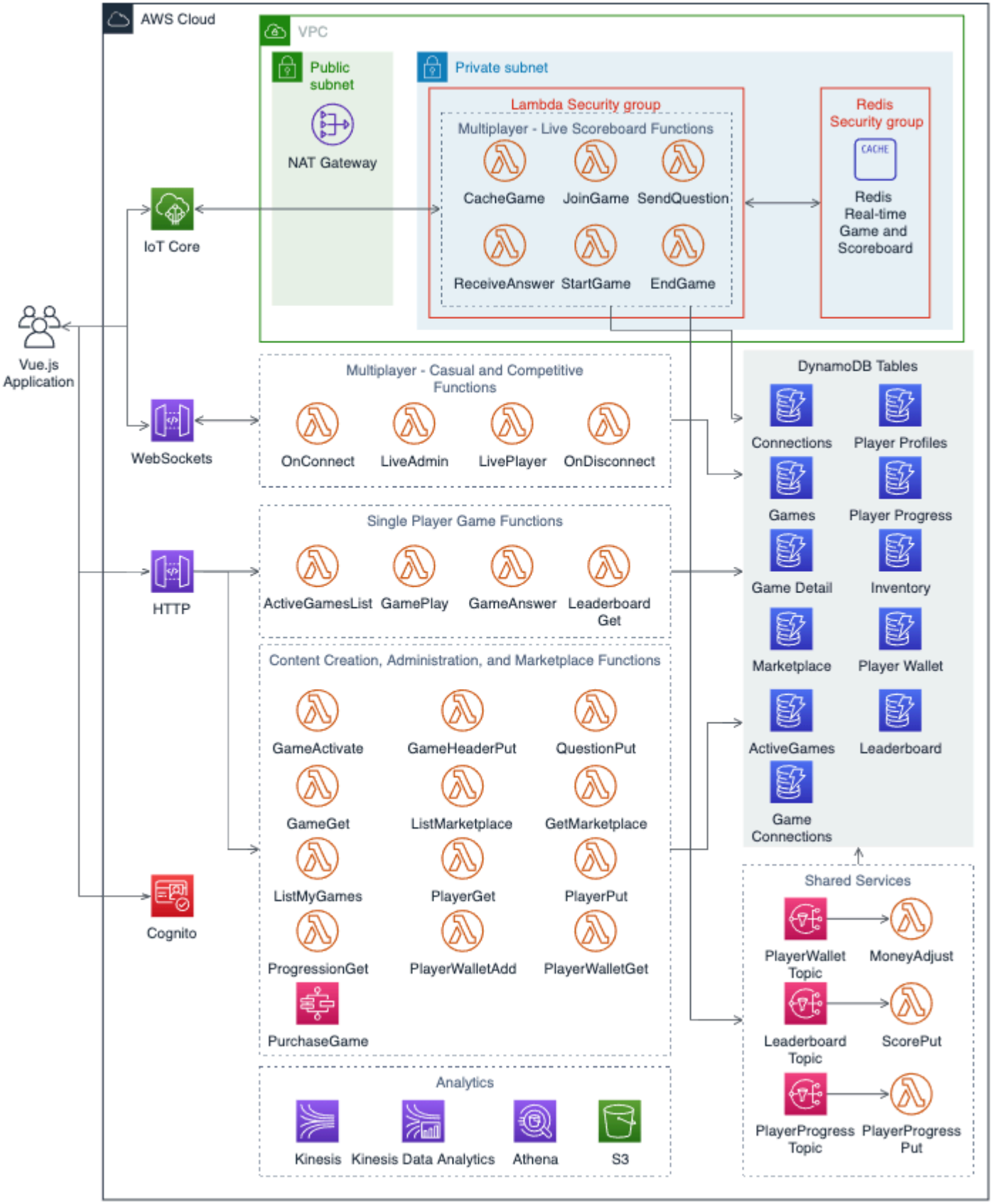
CloudFront pode ser usado para distribuir o software cliente do jogo para seus jogadores, bem como os ativos do jogo e outros conteúdos.

- AWS O Global Accelerator fornece aceleração de tráfego e controles personalizáveis para rotear o tráfego dos clientes de jogos do jogador para seus balanceadores de carga, bem como rotear o tráfego entre regiões para fins de multirregião e de failover. Os nomes de domínio personalizados para o back-end REST do seu jogo APIs são configurados no Amazon Route 53 para direcionar o tráfego para os endpoints do Global Accelerator. Não mostrado no diagrama,AWS Shield Advanced pode fornecer mitigação adicional de DDo S para seu acelerador e back-end de jogos.
- O NAT Gateway e o Application Load Balancer são implantados em sub-redes públicas em cada uma das zonas de disponibilidade usadas pelo back-end do jogo para fornecer alta disponibilidade à região.AWS WAF é implantado no Application Load Balancer para fornecer filtragem de tráfego da web de camada 7.
- O back-end do jogo é hospedado como uma coleção de microsserviços individuais baseados em contêineres implantados em um cluster Amazon EKS em sub-redes privadas espalhadas por zonas de disponibilidade para maior resiliência. O escalonamento automático ajusta dinamicamente a capacidade dos serviços e dos nós do cluster com base na utilização de recursos, que normalmente está correlacionada com a demanda dos jogadores. Enquanto o autoescalador de cluster ajusta automaticamente o número de nós no cluster, o escalonador automático horizontal de pods dimensiona automaticamente os pods implantados no cluster.
- Os dados do jogo e do jogador são armazenados em bancos de dados e caches de back-end que são implantados em sub-redes privadas em todas as zonas de disponibilidade com replicação entre os nós primário e de réplica. O Amazon Aurora é uma escolha popular para casos de uso como perfis de jogadores, direitos e compras no jogo, que podem ter requisitos de consulta mais complexos e podem se beneficiar dos recursos de modelagem de dados relacionais do MySQL e do PostgreSQL. ElastiCache A Amazon é útil para criar tabelas de classificação de alto desempenho, pub/sub enviar mensagens e armazenar em cache dados acessados com frequência para reduzir a latência e a carga nos bancos de dados. O Amazon DynamoDB é um armazenamento de dados NoSQL totalmente gerenciado que é ideal para padrões de acesso imprevisíveis e a capacidade de escalar para uma taxa de transferência praticamente ilimitada para casos de uso, como dados de estado de jogadores e jogos, dados de sessão, estoques de inventário e itens, ou casos de uso em que você deseja um banco de dados global com sobrecarga mínima.
- Os fluxos de trabalho de processamento assíncrono devem ser usados para realizar trabalhos que podem ser feitos em segundo plano, como atualizar tabelas de classificação ou enviar solicitações de amizade. Configure o back-end do seu jogo para colocar esse tipo de trabalho

nas filas do Amazon SQS para escalar à medida que seu jogo cresce ou considere usar tópicos do Amazon SNS para distribuir o trabalho entre várias filas de aplicativos de consumidores para processamento paralelo. Use as funções do AWS Lambda para realizar o processamento de forma orientada por eventos para reduzir seus custos de infraestrutura de computação e a sobrecarga de gerenciamento. Para fluxos de trabalho de longa duração ou que exigem coordenação de tarefas com várias etapas, considere orquestrar todo o fluxo de trabalho usando.AWS Step Functions A Amazon EventBridge pode ser usada para iniciar funções para responder a eventos AWS de serviços e aplicativos personalizados.

Arquitetura de back-end de jogos baseada em servidor

Muitos desenvolvedores de jogos não querem gerenciar a infraestrutura e, em vez disso, preferem criar seus jogos usando tecnologias que lhes permitam se concentrar no software. Uma arquitetura sem servidor é recomendada nesse cenário porque permite criar e liberar recursos mais rapidamente e com menos sobrecarga operacional. As arquiteturas sem servidor são projetadas usando serviços em nuvem que podem ser escalados dinamicamente com base na demanda, sem a necessidade de configurar, gerenciar e escalar servidores. A arquitetura de referência a seguir ilustra como criar um jogo usando uma arquitetura sem servidor.


AWS Cloud
VPC
Public subnet
NAT Gateway
Private subnet
Lambda Security group
Multiplayer - Live Scoreboard Functions
CacheGame
JoinGame
SendQuestion
ReceiveAnswer
StartGame
EndGame
Redis Security group
CACHE
Redis Real-time Game and Scoreboard
IoT Core
Vue.js Application
WebSockets
HTTP
Cognito
Multiplayer - Casual and Competitive Functions
OnConnect
LiveAdmin
LivePlayer
OnDisconnect
Single Player Game Functions
ActiveGamesList
GamePlay
GameAnswer
Leaderboard Get
Content Creation, Administration, and Marketplace Functions
GameActivate
GameHeaderPut
QuestionPut
GameGet
ListMarketplace
GetMarketplace
ListMyGames
PlayerGet
PlayerPut
ProgressionGet
PlayerWalletAdd
PlayerWalletGet
PurchaseGame
Analytics
Kinesis
Kinesis Data Analytics
Athena
S3
DynamoDB Tables
Connections
Player Profiles
Games
Player Progress
Game Detail
Inventory
Marketplace
Player Wallet
ActiveGames
Leaderboard
Game Connections
Shared Services
PlayerWallet Topic
MoneyAdjust
Leaderboard Topic
ScorePut
PlayerProgress Topic
PlayerProgress Put

Arquitetura de referência de back-end de jogos baseada em servidor

Essa arquitetura de referência ilustra um jogo de perguntas e respostas baseado na web que fornece recursos para um jogador e vários jogadores.

- Autenticação do jogador: os jogadores se autenticam usando o Amazon Cognito, que fornece autenticação segura com um diretório de usuários para o gerenciamento da identidade do jogador.
- Lógica do jogo como funções sem servidor: os recursos do jogo e a lógica de negócios de back-end são executados como AWS Lambda funções que são iniciadas em resposta a eventos, o que mantém os custos baixos porque você paga apenas quando a função é executada. O Lambda oferece a flexibilidade de escrever cada recurso do jogo como um microsserviço separado usando uma linguagem de programação de sua escolha. Por exemplo, você pode optar por desenvolver funções do.NET Lambda se tiver experiência no uso de C# para criar jogos Unity, ou pode optar por desenvolver funções Lambda do Node.js se desejar programar um front-end e um back-end para um jogo baseado na web em ambos. JavaScript
- Armazenamento de dados NoSQL para dados de jogos e jogadores: use o DynamoDB para armazenar seus dados de jogadores e jogos, pois ele foi criado especificamente para armazenar grandes quantidades de dados de microsserviços. Conforme ilustrado nessa arquitetura, é uma prática recomendada usar armazenamentos de dados separados para as necessidades de armazenamento de dados de cada recurso do jogo, o que facilita o monitoramento e o gerenciamento dos recursos de forma independente. Isso também cria limites de separação se a propriedade de um recurso ou serviço mudar em sua equipe. Nessa arquitetura de referência, as tabelas do DynamoDB são usadas para armazenar dados como estado da conexão, detalhes do jogo, progresso do jogador e informações da tabela de classificação.
- Jogabilidade para um jogador: os recursos para um jogador permitem que os jogadores realizem ações como selecionar e jogar um jogo e visualizar a tabela de classificação. Esses recursos são implementados como serviços de RESTful back-end hospedados com uma API HTTP do Amazon API Gateway que invoca a função Lambda apropriada para obter e definir dados nas tabelas do DynamoDB. Quando a jogabilidade termina, o back-end também envia notificações aos tópicos do Amazon SNS que iniciam as funções do Lambda de forma assíncrona para armazenar o progresso e as estatísticas do jogador.
- Jogabilidade multijogador: os recursos do jogo multijogador exigem que os jogadores possam interagir com o jogo para se point-to-point comunicar, bem como transmitir e receber atualizações de outros jogadores conectados. Uma WebSockets implementação é adequada para point-to-point comunicação em um jogo leve, como curiosidades. Os jogadores podem estabelecer uma WebSockets conexão com o Amazon API Gateway WebSockets, que gerencia a conexão e só

invoca as funções do Lambda quando há mensagens para enviar ou receber para um jogador. Para casos de uso em que a one-to-many comunicação é necessária entre jogadores, AWS IoT Core fornece suporte para o uso WebSockets de mensagens por meio do MQTT, o que permite que os clientes se inscrevam em tópicos e atuem de acordo com as mensagens recebidas. Nessa arquitetura, WebSockets mais de MQTT são usados para dar suporte a casos de uso, como transmitir atualizações ao vivo no jogo e fazer perguntas aos jogadores conectados. Como alternativa AWS IoT, você pode escolher Redis Pub/Sub para entrega de mensagens ou Redis Streams se precisar de retenção de mensagens.

- Use funções Lambda habilitadas para VPC para acessar recursos em suas sub-redes privadas: configure funções Lambda habilitadas para VPC para acessar recursos nas sub-redes privadas da sua VPC, como a ElastiCache Amazon, que é usada para reduzir os tempos de consulta para conjuntos de dados de baixa latência, como tabelas de classificação ativas.

Para obter mais informações, consulte Orientações para hospedagem personalizada de back-end de jogos em AWS.

Desenvolvimento de jogos em nuvem (CGD)

O desenvolvimento de jogos na nuvem (CGD) se refere à infraestrutura e às ferramentas necessárias para o ciclo de vida de desenvolvimento de jogos criar, testar e desenvolver um jogo. O desenvolvimento de jogos é colaborativo entre os usuários e os requisitos de infraestrutura mudam frequentemente ao longo dos estágios de desenvolvimento.

Muitos desenvolvedores de jogos estão adotando equipes de desenvolvimento remotas e distribuídas globalmente, o que requer tecnologia que suporte esse tipo de desenvolvimento. Os desenvolvedores de jogos podem hospedar todos ou parte desses ambientes AWS e usar a disponibilidade global do Regiões da AWS para colocar os recursos mais perto dos usuários e gerenciar seus ambientes de desenvolvimento de forma mais econômica, escalando a computação e o armazenamento conforme necessário.

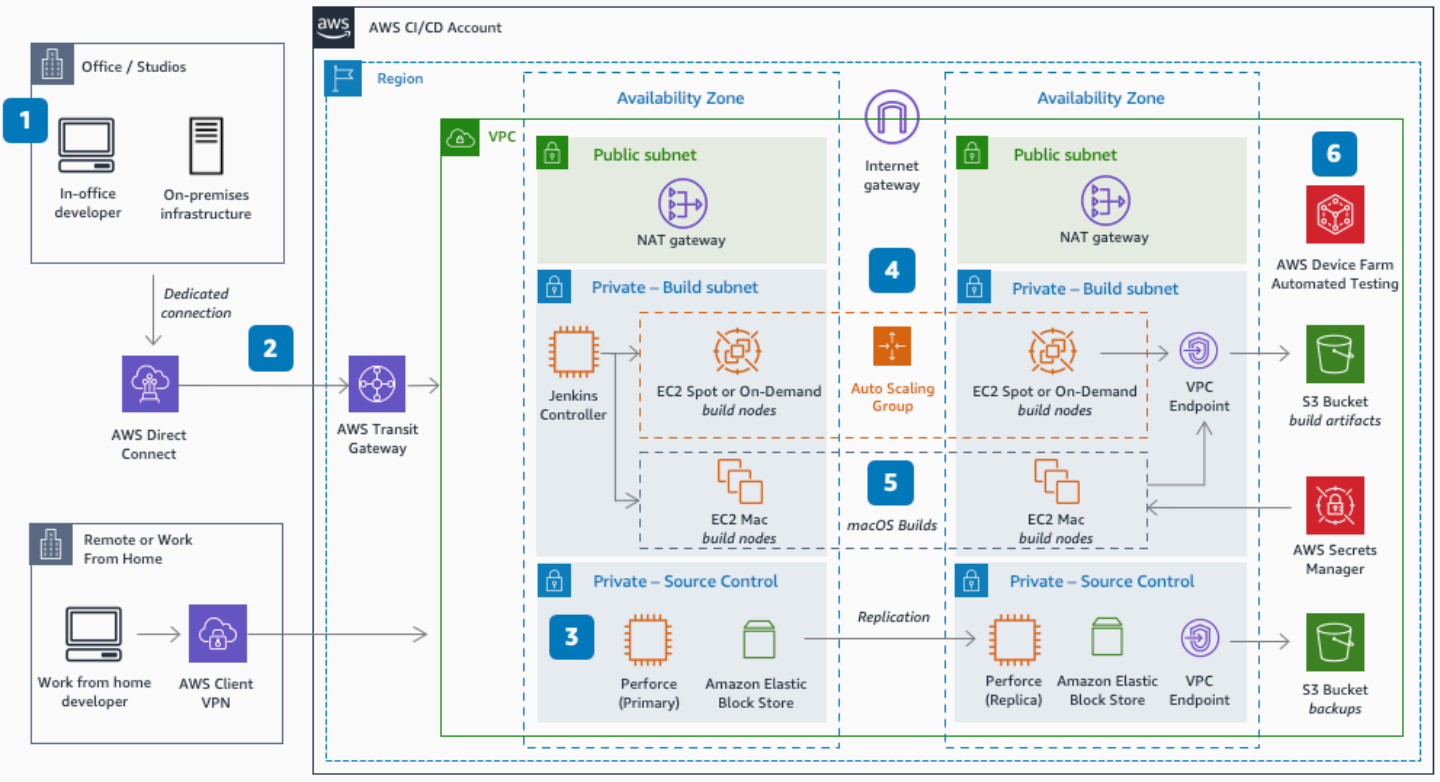
Os ambientes podem variar de acordo com as necessidades dos desenvolvedores de jogos, mas geralmente incluem estações de trabalho de desenvolvedores para artistas, designers, engenheiros, testadores de controle de qualidade, prestadores de serviços e outros funcionários realizarem seu trabalho. Esses ambientes geralmente também incluem um build farm composto por repositórios de código-fonte para que os usuários verifiquem suas alterações e a CI/CD infraestrutura para criar, empacotar e testar os artefatos desenvolvidos.

Essas arquiteturas de produção de jogos têm as seguintes características:

- Os usuários devem poder acessar uma estação de trabalho virtual por meio de um navegador da web ou de um cliente de desktop local, como o Amazon DCV, que fornece uma sessão de streaming de baixa latência para acessar o mesmo software e as mesmas ferramentas aos quais teriam acesso se estivessem trabalhando em uma máquina em um escritório ou estúdio de desenvolvimento. Essas estações de trabalho virtuais, normalmente um servidor baseado em nuvem, devem permitir que um usuário colabore e trabalhe em seus projetos inteiramente em um ambiente de nuvem por meio de uma LAN ou WAN. Quando os usuários não estão usando ativamente as máquinas, seu trabalho deve ser copiado para um armazenamento em nuvem durável, por exemplo, um repositório de controle de origem ou sistema de arquivos, como o Amazon Elastic File System (EFS) e o Amazon FSx, e a máquina deve ser desligada para reduzir custos.
- Os repositórios de controle de origem, como o Perforce, devem ser projetados com alta disponibilidade usando a replicação entre zonas de disponibilidade ou entre ambientes locais com backups armazenados em armazenamento em nuvem, como o Amazon S3. Por exemplo, um servidor Perforce baseado em nuvem deve incluir um servidor de confirmação primário hospedado em uma zona de disponibilidade com replicação para um servidor em espera hospedado em outra zona de disponibilidade na mesma região.
- Os recursos de build farm de desenvolvimento de jogos devem ser projetados com escalabilidade automática para que os recursos computacionais sejam provisionados conforme necessário, e as instâncias EC2 spot devem ser usadas para reduzir os custos incorridos ao expandir o número de servidores necessários para as compilações.

Desenvolvimento de jogos na nuvem: CI/CD

CI/CD a infraestrutura é importante ao desenvolver jogos, independentemente do tamanho da equipe, para melhorar os tempos de iteração, criar confiabilidade, implantação eficiente e melhor controle sobre o processo de desenvolvimento e lançamento para oferecer uma experiência de jogo de alta qualidade aos jogadores. Um CI/CD pipeline de desenvolvimento de jogos geralmente é composto por servidores e armazenamento de controle de origem altamente disponíveis, recursos computacionais para executar suas compilações e software para realizar testes automatizados, além da conectividade de rede adequada de suas máquinas de desenvolvimento. A arquitetura de referência a seguir demonstra como transferir compilações de jogos de ambientes de desenvolvimento de jogos remotos ou locais para o, a fim de ajudar os desenvolvedores a Nuvem AWS migrar ou construir novas fazendas de construção.



Transfira compilações de jogos para a nuvem

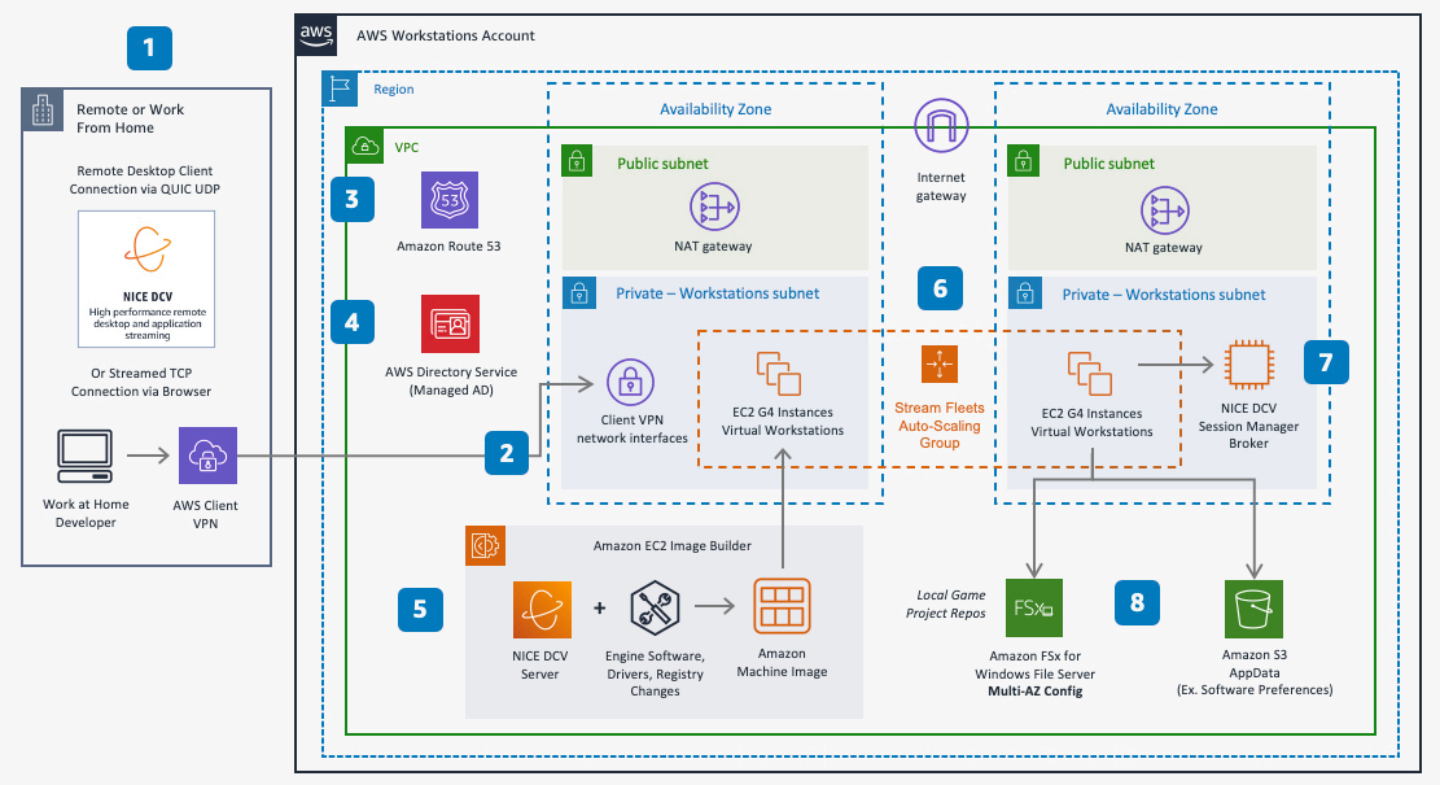
1. AWS Direct Connect fornece uma conexão dedicada, privada e de baixa latência AWS para desenvolvedores internos. Desenvolvedores remotos usam tecnologias de confiança zero Acesso Verificado pela AWS, como redes privadas virtuais (VPN), como AWS Client VPN.
2. AWS Transit Gateway simplifica o gerenciamento de rede para conectividade entre VPCs e a partir de redes locais.
3. O Perforce gerencia o controle de origem e versão (CI) apoiado pelo armazenamento do Amazon EBS para dados persistentes e de acesso rápido. O Perforce Helix Core está disponível no.AWS Marketplace
4. Os commits iniciam uma compilação (CD) no Jenkins quando os desenvolvedores enviam alterações para o Perforce vinculadas a uma ramificação. O Perforce inicia POST, uma carga JSON para Jenkins. O controlador Jenkins chama comandos CLI headless do engine para executar e paralelizar o processo de criação em nós efêmeros do Docker (como Amazon Spot Instances ou Amazon On-Demand Instances). EC2 EC2 Os desenvolvedores podem aumentar a disponibilidade usando dois controladores Jenkins, um em cada zona de disponibilidade, atrás de um balanceador de carga. Para alguns mecanismos de jogos, os desenvolvedores podem precisar de uma infraestrutura de licenciamento adicional configurada em sub-redes adicionais para vender licenças para o contexto de compilação sempre que uma compilação simultânea for executada.

5. A parte do Xcode das compilações do iOS é transferida para as instâncias do EC2 Amazon Mac para assinar, criar e exportar o arquivo.IPA, dividindo o processo e reduzindo os tempos de compilação.AWS Secrets Manager contém perfis de provisionamento, chaves privadas e certificados.
6. Os artefatos de construção são entregues ao Amazon S3, que envia notificações de sucesso ou falha. AWS Device Farm permite testes automatizados para dispositivos móveis.

Desenvolvimento de jogos na nuvem: estações de trabalho

O desenvolvimento de jogos geralmente envolve equipes distribuídas trabalhando remotamente de vários locais, exigindo acesso à infraestrutura compartilhada e a capacidade de apoiar o desenvolvimento colaborativo e geograficamente disperso. Essa tendência de um processo de desenvolvimento de jogos mais distribuído, incluindo o uso de work-for-hire estúdios e trabalho remoto, exige a implementação de tecnologias e fluxos de trabalho robustos para facilitar a produtividade, o conhecimento compartilhado e as práticas de desenvolvimento ágil.

A arquitetura de referência a seguir demonstra como usar AWS para hospedar estações de trabalho remotas de desenvolvimento de jogos usando o protocolo Amazon DCV.



Transmita o desenvolvimento de jogos de qualquer lugar com o Amazon DCV

1. O Amazon DCV é um protocolo de streaming compatível com streaming de 4K e 60 FPS. Os desenvolvedores que usam um navegador se conectam por meio de conexões TCP, enquanto os clientes de desktop podem usar o QUIC UDP pela porta 8443 para aumentar o desempenho.
2. Os desenvolvedores usam o AWS Client VPN para uma conexão segura com interfaces de rede nas sub-redes da estação de trabalho com tradução de endereço de rede de origem (SNAT).
3. O Amazon Route 53 fornece DNS privado para os recursos na VPC, bem como encaminhamento de DNS de entrada e saída.
4. Directory Service fornece o Microsoft Active Directory gerenciado para permitir o armazenamento local de projetos de jogos mapeado para usuários individuais.
5. As estações de trabalho são criadas usando uma Amazon Machine Image (AMI) criada com o Image Builder. As imagens incluem o Amazon DCV Server, software para desenvolvedores, alterações no registro e drivers como drivers de jogos ou drivers periféricos da NVIDIA. AWS Marketplace inclui o AMIs uso comum para estações de trabalho.
6. Frotas de estações de trabalho usam tipos de EC2 instância gráficos da Amazon que fornecem GPUs e são escalados usando grupos de Auto Scaling. EC2
7. O Amazon DCV Session Manager Broker permite o gerenciamento de sessões do Amazon DCV.
8. O armazenamento local de arquivos dos projetos é hospedado no Amazon FSx for Windows File Server. Os desenvolvedores se comprometem com um pipeline de CI/CD separado, passando do armazenamento local para o controle de origem.

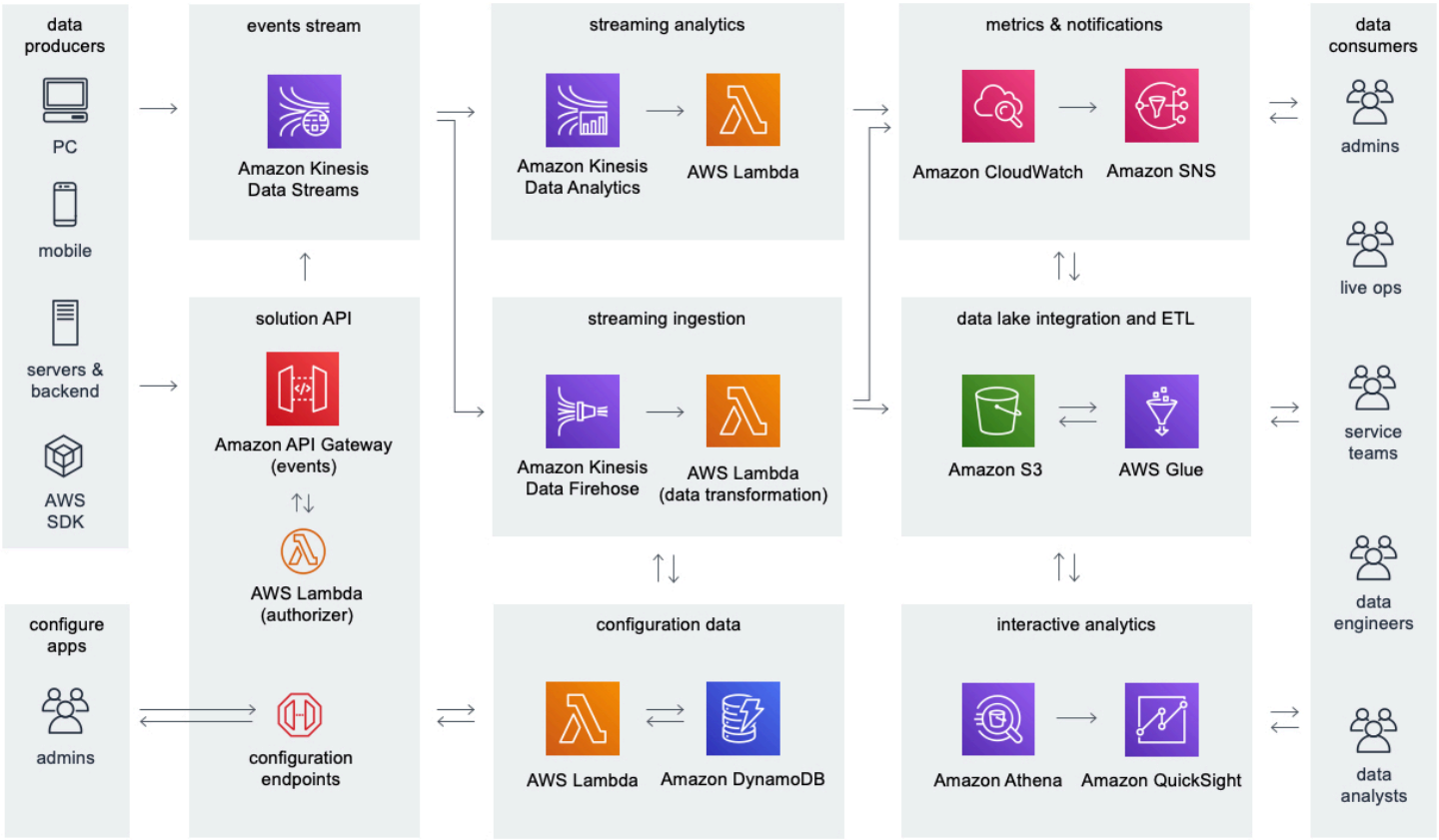
Game Analytics Pipeline (Pipeline de análises de jogos)

Os desenvolvedores de jogos estão cada vez mais procurando maneiras de entender melhor o comportamento dos jogadores para que possam melhorar a experiência de jogo para reter e aumentar sua base de jogadores. A análise de jogos representa a infraestrutura técnica e os processos necessários para entender e analisar os dados gerados pelo jogo e pelos serviços relacionados. Isso normalmente requer o uso de uma arquitetura de pipeline de análise que possa suportar esse end-to-end processo, como a implementação da solução Game Analytics Pipeline.

As arquiteturas de análise de jogos têm as seguintes características:

- As fontes de dados enviam dados em um formato comum, como JSON, e geralmente incluem servidores e serviços de back-end de jogos, bem como clientes de jogos, incluindo PC, dispositivos móveis e consoles de jogos.

- Um pipeline de análise de jogos automatiza todo o fluxo de trabalho de ingestão e armazenamento de dados brutos e processamento em formatos de saída utilizáveis para que possam ser analisados de forma eficiente e econômica pelos consumidores de dados, como usuários finais e aplicativos de análise.
- Os pipelines de análise de jogos fornecem suporte para ingerir e processar grandes volumes de dados em tempo real para escalar à medida que o jogo cresce.
- Forneça suporte para casos de uso de relatórios em tempo real e em lote. Por exemplo, painéis e alertas em tempo real são normalmente usados por equipes de operações ao vivo para monitorar a infraestrutura do jogo e o comportamento dos jogadores para detectar problemas. As equipes de analistas de dados geralmente contam com relatórios em lote e conforme necessário para entender as tendências ao longo do tempo.



Pipeline de análise de jogos sem servidor para telemetria de jogabilidade

Os dados do jogo são ingeridos de clientes de jogos, servidores de jogos e outros aplicativos. Os dados de streaming são ingeridos no Amazon S3 para integração de data lake e análise interativa. A análise de streaming processa eventos em tempo real e gera métricas. Os consumidores de dados analisam dados de métricas na Amazon CloudWatch e eventos brutos no Amazon S3.

- API da solução e dados de configuração: use o Amazon API Gateway para fornecer uma API REST para administrar seu pipeline de análise de jogos e armazenar dados de configuração no Amazon DynamoDB usando funções Lambda. Você pode criar um portal interno sobre essa API ou uma interface de linha de comando personalizada para administração. Uma API REST também fornece autenticação de servidor para ingerir dados de jogabilidade de fontes de dados e encaminhar os dados de telemetria para o Amazon Kinesis Data Streams para processamento e ingestão em tempo real no armazenamento.
- Stream de eventos: o Amazon Kinesis Data Streams captura dados de streaming do seu jogo e permite o processamento de dados em tempo real pelo Amazon Data Firehose e pelo Amazon Managed Service para Apache Flink.
- Análise de streaming: o Managed Service for Apache Flink analisa dados de eventos de streaming do Kinesis Data Streams e pode gerar métricas e alertas personalizados que são publicados usando funções Lambda. CloudWatch
- Métricas e notificações: use CloudWatch a Amazon para monitorar as métricas, os registros e os alarmes da sua solução. Use o Amazon SNS para enviar notificações para engenheiros de plantão e outros consumidores de dados.
- Ingestão de streaming: use o Firehose para consumir seus dados de streaming do Kinesis Data Streams e entregá-los ao seu data lake no Amazon S3 para armazenamento de longo prazo, transformação e integração com outros dados.
- Integração de data lake e ETL: use AWS Glue para fluxos de trabalho de processamento de ETL e para organizar seus metadados no AWS AWS Glue Data Catalog, que fornece a base para um data lake para integração com ferramentas de análise flexíveis.
- Análise interativa: os usuários finais podem usar o Amazon Athena para realizar consultas interativas ad-hoc nos conjuntos de dados armazenados no Amazon S3, e o Quick Suite pode ser usado para criar painéis.

Consulte o Game Analytics Pipeline para obter uma implementação de referência automatizada de um pipeline de análise que pode ser implantado em sua conta usando AWS CloudFormation.

Definições

O AWS Well-Architected Framework é baseado em seis pilares: excelência operacional, segurança, confiabilidade, eficiência de desempenho, otimização de custos e sustentabilidade. AWS fornece vários componentes principais que permitem projetar state-of-the-art arquiteturas para suas cargas de trabalho de jogos. Nesta seção, apresentaremos uma visão geral das principais definições.

Para os fins deste paper, uma arquitetura de jogo engloba a infraestrutura técnica de back-end necessária para criar e operar um jogo. Alguns jogos podem não ter recursos sociais, multijogador ou outros recursos online e podem não exigir o uso de certos aspectos da infraestrutura técnica de back-end descritos neste paper. Para uma discussão detalhada dos diferentes tipos de cargas de trabalho que são frequentemente implantadas para dar suporte a uma arquitetura de jogo, consulte Cenários.

A Nuvem AWS infraestrutura é construída em torno de regiões e zonas de disponibilidade.

- Uma região é um local físico no mundo em que temos várias zonas de disponibilidade.
- As zonas de disponibilidade consistem em um ou mais data centers discretos, cada um com energia, rede e conectividade redundantes, alojados em instalações separadas.

Dependendo das características do seu jogo, talvez você queira implantar certos componentes da sua arquitetura de jogo em várias regiões por motivos como melhorar o desempenho dos jogadores ou oferecer experiências personalizadas aos jogadores, dependendo da localização.

Há muitos tipos diferentes de jogos, e a infraestrutura técnica de back-end necessária para suportar um jogo varia de acordo com o tipo de jogo que está sendo desenvolvido. Por exemplo, tipos populares de jogos podem incluir jogos de tiro em primeira pessoa (FPS), jogos de RPG (RPG), jogos online multijogador massivo (MMOG), battle royales (BR), jogos de esportes, jogos de quebra-cabeça e muito mais. Também existem diferentes modos de interação de jogo que influenciam a arquitetura do jogo, como jogo por turnos e simultâneo, com diferentes características de desempenho.

Os jogos são desenvolvidos para serem jogados em um ou mais sistemas de jogos, incluindo desktop, web, dispositivos móveis, consoles e modos de interação mais recentes, como realidade aumentada (AR), realidade virtual (VR) e soluções de streaming de jogos. Os jogos geralmente oferecem suporte à jogabilidade entre sistemas, o que significa que os jogadores podem salvar sua

progressão no jogo e continuar a jogar em outros sistemas, bem como iniciar sessões de jogo com jogadores em outros sistemas.

A monetização de videogames permite que os editores de jogos gerem receita usando estratégias diferentes, como publicidade, compras de jogos digitais e de varejo, compras de conteúdo para download (DLC) no jogo, conhecidas como microtransações, e por meio de assinaturas pagas obrigatórias para jogar o jogo. Alguns dos principais indicadores de desempenho (KPIs) mais comuns na indústria de jogos incluem:

- Usuários ativos diariamente (DAU)
- Usuários ativos mensalmente (MAU)
- Usuários simultâneos (CCU)
- Duração da sessão
- Custo por instalação (CPI)
- Valor da vida útil do jogador (LTV)
- Receita média por usuário (ARPU)

Sistema de jogo

Os videogames são desenvolvidos para serem jogados em um sistema de jogo que fornece controles de entrada ao cliente, gráficos, software cliente (conhecido como cliente do jogo) e hardware e, em alguns casos, recursos exclusivos do sistema para suportar a jogabilidade.

Os sistemas de jogos geralmente são separados nas seguintes categorias:

- Consoles: sistemas de entretenimento desenvolvidos especificamente para jogos, incluindo exemplos populares como Sony, PlayStation Microsoft Xbox e Nintendo Switch. Os consoles oferecem a capacidade de jogar jogos instalando conteúdo físico ou distribuído digitalmente no hardware do console fabricado pelo fornecedor do sistema de jogos. Nessa definição, um console pode ser portátil ou estacionário e destinado a ser usado em um cenário de entretenimento doméstico.
- Computador pessoal (PC): jogos que são jogados usando um software de computador instalado em uma máquina cliente que pode ser personalizado pelo jogador. Por esse motivo, os jogos para PC são populares entre os jogadores devido à flexibilidade e ao controle que oferecem.

- Web: jogos projetados para serem jogados usando um navegador da Web e que geralmente oferecem o benefício de permitir que um jogador acesse o jogo independentemente do sistema operacional.
- Celular: jogos desenvolvidos para serem jogados em um telefone celular, geralmente um sistema operacional de smartphone. Os jogos para celular geralmente são baixados de uma loja de aplicativos digitais e instalados no telefone.

Além dos sistemas mencionados anteriormente, também existem sistemas emergentes que ainda são relativamente novos e estão crescendo e têm uma participação de mercado muito menor em comparação com os sistemas mais predominantes. Exemplos de sistemas de jogos nessa categoria incluem AR, VR e streaming de jogos, às vezes chamados de jogos em nuvem.

O streaming de jogos envolve renderizar a jogabilidade na nuvem e transmitir para um thin client, normalmente um navegador. O streaming de jogos permite que um jogador jogue um jogo totalmente hospedado remotamente, normalmente na nuvem por um provedor de serviços de streaming de jogos. No streaming de jogos, o jogador se conecta a um jogo baseado em nuvem por meio de um navegador ou de um thin client fornecido pelo provedor de serviços de jogos em nuvem (sistema de jogos).

Servidor de jogos

Os servidores de jogos representam um dos aspectos mais importantes da infraestrutura computacional do seu jogo. Os servidores de jogos, às vezes chamados de servidores de jogos dedicados, são usados no desenvolvimento de um jogo multijogador ou quando é necessário o processamento autoritário de eventos de jogo pelo servidor. O servidor do jogo está no centro da arquitetura do jogo, servindo como o local onde a lógica central é executada, o que inclui o gerenciamento do jogador e do estado do jogo, bem como o gerenciamento das interações entre os clientes do jogo conectados e o servidor do jogo. O servidor de jogos geralmente é um dos aspectos mais sensíveis ao desempenho de uma arquitetura de jogo porque é responsável por processar as entradas do cliente de jogo de um jogador e distribuí-las adequadamente para outros jogadores conectados em tempo real. Um servidor de jogo com baixo desempenho afeta o desempenho geral da experiência de jogo. Portanto, você deve otimizar o desempenho do servidor de jogos e fornecer capacidade suficiente, especialmente no lançamento do jogo ou nos períodos de pico do jogo.

Para os fins deste documento, um servidor de jogos ou instância de servidor de jogos se refere à computação, como uma máquina virtual, que hospeda um ou mais processos do servidor de jogos. Um processo de servidor de jogos representa uma única instância da sua compilação de servidor

de jogos hospedando uma sessão de jogo, que é uma instância do seu jogo em execução à qual os jogadores podem se conectar por meio de uma sessão de jogador. Por esse motivo, geralmente nos referimos ao processo do servidor do jogo ou à sessão do jogo de forma intercambiável devido à relação implícita de um para um entre uma sessão de jogo e o processo do servidor de jogo que a hospeda. Em AWS, há várias opções de computação para hospedar servidores de jogos, que fornecem acesso à capacidade escalável baseada em nuvem por meio do provisionamento elástico de recursos.

EC2 A Amazon fornece servidores virtuais baseados em nuvem, conhecidos como instâncias, com suporte para várias versões do Linux e do Windows. Você pode criar instâncias e gerenciá-las diretamente, como outro servidor ou máquina virtual. Normalmente, vários processos de servidor de jogos são implantados em uma instância para melhorar a eficiência e reduzir custos. EC2 A Amazon é uma boa opção para servidores de jogos se você deseja o máximo controle sobre a infraestrutura computacional.

GameLift A Amazon fornece uma solução totalmente gerenciada para hospedagem de servidores de jogos dedicados na nuvem, bem como recursos adicionais, como matchmaking com GameLift FlexMatch. GameLiftfornece uma camada de abstração sobre EC2 a Amazon para simplificar o gerenciamento de servidores de jogos e está disponível na maioria para que você Regiões da AWS possa hospedar servidores de jogos perto dos jogadores para reduzir a latência, obter alta disponibilidade e reduzir significativamente os custos usando instâncias spot. Embora GameLift possa ser integrado aos back-ends de jogos existentes, é especialmente útil para desenvolvedores de jogos que não desejam desenvolver suas próprias soluções de gerenciamento de servidores e matchmaking e preferem uma solução que seja gerenciada AWS e possa ser escalada à medida que o jogo cresce.

O Amazon Elastic Container Service (Amazon ECS) é um serviço de orquestração de contêineres totalmente gerenciado que você pode usar para executar contêineres baseados em Docker. Você também pode usar o Amazon Elastic Kubernetes Service (Amazon EKS) para executar contêineres baseados em Docker criados usando o Kubernetes. O uso de tecnologias de contêiner, como as fornecidas pelo Amazon ECS e pelo Amazon EKS, pode ajudá-lo a melhorar a utilização da computação ao agrupar com eficiência muitos processos de servidores de jogos ou outras instâncias de aplicativos de jogos em uma instância. EC2

O uso de contêineres também pode melhorar a produtividade do desenvolvedor hospedando aplicativos usando o mesmo tempo de execução operacional de imagem Docker usado pelos desenvolvedores em suas máquinas locais durante o desenvolvimento. Você pode reduzir ainda mais a sobrecarga operacional usando AWS Fargate, que é uma solução de computação sem

servidor para executar contêineres e é compatível com o Amazon EKS e o Amazon ECS. O Fargate é mais adequado para casos de uso em que você deseja executar servidores de jogos em contêineres sem a responsabilidade de operar as instâncias subjacentes nas quais os contêineres são executados.

Você pode usar AWS Outposts para executar AWS infraestrutura e serviços em um data center ou instalação local, o que pode permitir que os jogos sejam executados em ambientes locais e AWS usar a mesma infraestrutura para apoiar uma estratégia de adoção da nuvem híbrida. AWS As Zonas Locais servem como extensões Regiões da AWS que permitem que seus servidores de jogos e outras cargas de trabalho sensíveis à latência funcionem mais perto de seus jogadores ou equipes de desenvolvimento. Além disso, para reduzir a latência da rede global para seus servidores de jogos, você pode usar o AWS Global Accelerator para melhorar o desempenho do tráfego de jogadores em seus servidores de jogos.

AWS Lambda é um serviço de computação sem servidor que executa código sem provisionar ou gerenciar servidores, o que o torna útil para casos de uso assíncronos de servidores de jogos, como jogos baseados em turnos ou aqueles que têm requisitos computacionais leves, uma pequena base de código e onde a funcionalidade de jogabilidade pode ser projetada usando uma arquitetura de microsserviços sem estado. É importante ter em mente que as funções do Lambda são executadas de forma orientada por eventos, por solicitação, em vez de executar um processo de servidor de jogos de longa duração. O Lambda fornece a maior abstração em tempo de execução das opções deste paper porque o aplicativo subjacente está prontamente disponível para os desenvolvedores escolherem hospedar seu código.

Ao selecionar sua abordagem para hospedagem de servidores de jogos, considere vários requisitos, incluindo sobrecarga operacional, bases de código antigas, requisitos de desempenho e escala. EC2instâncias e contêineres são boas opções para bases de código legadas, pois exigem a menor alteração para serem migradas para a nuvem, e você pode usar EC2 instâncias para dedicar os recursos de uma instância de computação, enquanto os contêineres podem simplificar o gerenciamento e a alta utilização. As funções sem servidor oferecem o mais alto nível de abstração, que você pode usar para definir um código que só é executado em resposta a eventos, o que pode reduzir custos.

Clientes de jogo

O cliente do jogo representa o dispositivo de software e hardware que o jogador usa para jogar um jogo. O cliente do jogo fornece o software para traduzir as entradas do jogador em mensagens que são enviadas a um servidor para processamento e é responsável por lidar com as respostas

recebidas do servidor e renderizar saídas, como gráficos, para o jogador. Em jogos multijogador em rede em tempo real, o cliente do jogo geralmente mantém uma conexão de rede persistente com um servidor de jogo durante a sessão de jogo para reduzir a latência da rede e minimizar o tempo de processamento. No entanto, o cliente do jogo também pode interagir usando REST com um servidor de jogos ou serviços de back-end.

Sistema de mensagens

Normalmente, há três categorias principais de mensagens nos jogos:

- Mensagens de engajamento de jogadores direcionadas a um usuário específico ou grupo de usuários, como convites para jogos ou notificações push
- Mensagens em grupo entre jogadores, como bate-papo no jogo
- ervice-to-serviceMensagens S, como mensagens JSON usadas para integrar dois ou mais aplicativos

Uma estratégia comum para enviar e receber esses tipos de mensagens é usar padrões de arquitetura de processamento assíncrono e publicador-assinante. AWS fornece vários serviços que podem ajudá-lo a implementar mensagens em seu jogo.

- Amazon Simple Notification Service (SNS): serviço gerenciado para entrega de mensagens entre editores e assinantes usando um padrão de arquitetura. pub/sub Os editores enviam mensagens usando uma API para o Amazon SNS, que entrega as mensagens de forma assíncrona aos aplicativos assinantes e pode entregar notificações push diretamente para clientes móveis ou desktops com suporte para alguns dos serviços de notificação push mais usados. O Amazon SNS pode ser usado para notificações push para clientes, bem como para casos de uso service-to-service de mensagens.
- Amazon Simple Queue Service (SQS): um serviço de filas totalmente gerenciado que facilita a integração de servidores de jogos e seu jogo, independentemente da linguagem de programação usada em cada um. Muitas tarefas de jogos podem ser dissociadas e realizadas em segundo plano, como atualizar uma tabela de classificação ou valores de tempo de jogo em um banco de dados. Essa abordagem é uma forma eficaz de dissociar várias partes do jogo e escalar de forma independente os recursos voltados para o jogador do processamento de back-end.
- Amazon Managed Streaming for Apache Kafka (MSK): um serviço totalmente gerenciado que simplifica a criação de streaming de dados e aplicativos para produtores ou consumidores usando o Apache Kafka, uma solução popular de código aberto. O Kafka é normalmente usado para

ingerir e processar dados de streaming em tempo real e pode ser usado para enviar mensagens. service-to-service

- Amazon ElastiCache (Redis OSS): fornece um armazenamento de dados na memória totalmente gerenciado que inclui suporte ao popular pub/sub recurso do Redis, comumente usado para desenvolver aplicativos de sala de bate-papo e mensagens de alto desempenho. service-to-service O Redis também oferece suporte a tipos de dados avançados, como listas e conjuntos, para que os desenvolvedores possam usar o Redis para enfileiramento de alto desempenho.
- Amazon Pinpoint: fornece mensagens de engajamento do usuário por e-mail, SMS, voz e notificações push. Por exemplo, o Amazon Pinpoint pode ser usado para entregar mensagens de engajamento do usuário aos jogadores para convidá-los a voltar ao jogo e pode ser usado para casos de uso transacionais, como suporte a tokens de autenticação multifatorial, confirmações de pedidos e e-mails de redefinição de senha.

Operações de jogo ao vivo (Live Ops)

As operações de jogos ao vivo (Live Ops) são um estilo de gerenciamento e operações de jogos que trata um jogo como um serviço ao vivo e oferece continuamente novos recursos, atualizações, promoções, eventos no jogo e melhorias no jogo lançado para melhorar a experiência da comunidade de jogadores.

Tradicionalmente, os jogos eram entregues como produtos em vez de serviços, e novos conteúdos e recursos eram frequentemente incorporados em lançamentos ou sequências subsequentes, e não no produto lançado. Com uma abordagem Live Ops para gerenciamento de jogos, uma equipe de operações de jogos pode lançar um jogo e manter uma comunidade de jogadores engajada por meio de experimentação, promoções, eventos no jogo e inovação para manter os jogadores entretidos.

Embora essa abordagem tenha o benefício de desbloquear novas estratégias de engajamento de jogadores e fornecer fluxos de receita recorrentes, ela requer mais experiência operacional. Por exemplo, para implementar uma estratégia bem-sucedida de Live Ops, um desenvolvedor pode precisar se integrar aos serviços em nuvem ou operar sua própria infraestrutura técnica de back-end. Eles também precisam de uma forma eficaz de identificar e responder aos problemas que surgem no jogo ou na comunidade de jogadores, que podem impactar negativamente a experiência do jogador.

Excelência operacional

O pilar de excelência operacional se concentra nas melhores práticas para implantar e operar jogos baseados em nuvem em grande escala. É importante focar na excelência operacional para manter uma experiência positiva do jogador e implementar medidas preventivas para se preparar e se recuperar de problemas que afetam sua experiência.

Áreas de foco

- Princípios de design
- Operações ao vivo
- Estrutura da conta
- Implantações de jogos
- Monitoramento de saúde
- Testes de carga
- Otimização ao longo do tempo
- Recursos

Princípios de design

Além dos princípios de design do whitepaper Well-Architected Framework, os seguintes princípios de design podem ajudá-lo a alcançar a excelência operacional na criação e operação de jogos:

- Defina objetivos mensuráveis e alcançáveis para as equipes de operações de jogos e adapte-os conforme necessário: devido à natureza dos jogos baseada em acertos, é difícil determinar com antecedência quantos jogadores jogarão seu jogo quando ele for lançado ou quais expectativas os jogadores terão em relação às operações contínuas do jogo. É importante definir metas ambiciosas, mas alcançáveis, com as partes interessadas e criar uma abordagem que possa ser ampliada se o jogo exceder as projeções e reduzida, enquanto as equipes de desenvolvimento de jogos otimizam a experiência do jogador. Prepare-se e teste adequadamente com antecedência para atender a esses requisitos e alinhar suas partes interessadas comerciais e técnicas aos objetivos desejados para operar o jogo. Com as metas definidas, as equipes de jogo podem alcançar um equilíbrio adequado entre custo e desempenho durante o planejamento, projeto, provisionamento, teste, implantação e operação da infraestrutura de back-end do jogo.

- Use runbooks operacionais para planejar atividades de escalabilidade relacionadas a lançamentos de jogos e eventos especiais: as equipes de operações de jogos devem se coordenar com as partes interessadas da empresa para modelar projeções para o pico previsto de simultaneidade de jogadores em eventos e realizar um planejamento proativo para pré-escalar a capacidade da infraestrutura com antecedência. Devido à natureza flutuante do tráfego de jogadores durante os eventos, o planejamento prévio e as atividades de pré-escalonamento devem ampliar seus sistemas de escalonamento automatizado existentes para aumentar suas chances de sucesso durante um evento e verificar se você tem recursos suficientes para proporcionar uma experiência positiva aos jogadores. Implemente práticas de engenharia de desempenho para desenvolver uma linha de base de seus recursos e uma compreensão baseada em dados da capacidade do seu sistema, o que ajudará a orientar as atividades de pré-escalonamento e as configurações de escalabilidade automatizadas. Desenvolva runbooks operacionais para fornecer consistência no processo. Esse planejamento antecipado e a capacidade de resposta à demanda dos jogadores são especialmente essenciais para jogos de serviço ao vivo, que devem manter desempenho e infraestrutura confiáveis para apoiar uma base de jogadores ativa e engajada por um longo período de tempo.
- Estabeleça um modelo operacional para receber, investigar e responder às solicitações de suporte do jogador: após o lançamento, monitore os relatórios de reclamações e problemas com o jogo. Implemente sistemas apropriados para interagir com os jogadores de forma segura e eficaz para resolver adequadamente os problemas dos jogadores, como fóruns comunitários, mídias sociais, e-mail, sistemas de bilhetagem, call centers ou soluções automatizadas de chatbots. Isso é especialmente crucial para jogos de serviço ao vivo, que exigem comunicação contínua com a base de jogadores, capacidade de resposta ao feedback dos jogadores para atender às necessidades em evolução e manutenção de uma comunidade engajada por uma vida útil prolongada.

Operações ao vivo

GAMEOPS01: Como você define a estratégia de operações ao vivo (Live Ops) do seu jogo?

Desenvolva uma estratégia de operações ao vivo (Live Ops) para seu jogo com base em objetivos definidos e métricas de desempenho, em consulta com as partes interessadas da empresa.

Práticas recomendadas

GAMEOPS01-BP01 Use objetivos do jogo e métricas de desempenho comercial para desenvolver sua estratégia de operações ao vivo

Consulte as partes interessadas da empresa, como produtores de jogos e parceiros de publicação, para determinar os objetivos e as métricas de desempenho de um jogo. Isso pode ajudá-lo a desenvolver planos de como você gerenciará o jogo, incluindo a definição de janelas de manutenção, cronogramas de atualização de software e infraestrutura e metas de confiabilidade e capacidade de recuperação do sistema.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Essas métricas também podem ajudá-lo a determinar em qual estágio do ciclo de vida do seu jogo você deve incorporar um stream de operação ao vivo (Live Ops) para monitorar a integridade do jogo, coletar feedback direto do jogo e criar processos de lançamento simplificados e automatizados. Por exemplo, um novo jogo pode esperar até que uma determinada escala seja alcançada, medida pela contagem ativa de jogadores, receita ou outro conjunto de métricas, antes de montar uma equipe de operações ao vivo dedicada. Um estúdio de desenvolvimento de jogos estabelecido pode já ter experiência em operações ao vivo, talvez em outros jogos, então precisaria apenas integrar o novo jogo.

Etapas de implementação

- Você pode definir as metas para a simultaneidade de jogadores (CCU) e para os usuários ativos diários e mensais (DAU e MAU) que a infraestrutura do jogo deve ser capaz de suportar de forma eficaz, seus orçamentos de infraestrutura, metas financeiras e outras metas de desempenho, como a frequência de lançamento de conteúdo e recursos para aumentar o engajamento dos jogadores. Esses objetivos e métricas alimentam as decisões sobre o design do jogo, o gerenciamento de lançamentos, a observabilidade e o suporte necessários para operações eficientes.
- Seu jogo pode ter como objetivo lançar novas atualizações de conteúdo pelo menos uma vez por mês, sem tempo de inatividade durante o lançamento. Essas informações ajudam você a definir sua estratégia de implantação de lançamentos e coordenar o agendamento da manutenção

necessária, que pode exigir tempo de inatividade em outros momentos do mês, além de contribuir para seu SLA de disponibilidade.

Estrutura da conta

GAMEOPS02: Como você estrutura seus ambientes Contas da AWS para hospedar seus jogos?

Implemente uma estratégia de várias contas para isolar diferentes ambientes de jogo e aprimorar a segurança, a eficiência operacional e a escalabilidade. Use AWS Organizations para gerenciar hierarquias de contas, aplicar proteções às contas e aplicar políticas de tags e marcações nos recursos implantados.

Práticas recomendadas

- GAMEOPS02-BP01 Adote uma estratégia de várias contas para isolar diferentes jogos e aplicativos em suas próprias contas
- GAMEOPS02-BP02 Organize recursos de infraestrutura usando marcação de recursos

GAMEOPS02-BP01 Adote uma estratégia de várias contas para isolar diferentes jogos e aplicativos em suas próprias contas

Crie uma estrutura de contas que orientaria a implantação da infraestrutura de acordo com as necessidades operacionais, de segurança e de isolamento de cada ambiente. O isolamento do ambiente, restringindo o acesso a ele e permitindo que somente os AWS serviços necessários sejam usados nele, é essencial, com os ambientes de produção bloqueados, enquanto os ambientes de desenvolvimento e teste são tolerantes para permitir a experimentação. É altamente recomendável um maior isolamento dos principais subsistemas em cada ambiente e serviços comuns usados por vários ambientes para serem hospedados e gerenciados por conta própria Contas da AWS .

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Adote uma estratégia de várias contas AWS isolando os diferentes ambientes (como desenvolvimento, teste, preparação, produção e serviços compartilhados) de forma individual Contas

da AWS, o que reduz o escopo dos incidentes. Considere gerenciar centralmente sua Contas da AWS hierarquia AWS Organizations para simplificar ainda mais as operações, bem como definir e aplicar políticas em nível de conta e de unidade organizacional (nível de UO) de forma seletiva. Ao projetar uma OU e uma Conta da AWS estrutura adequadas que estejam alinhadas às suas necessidades de desenvolvimento e fluxo de trabalho de produção, você pode otimizar seus custos e aprimorar a escalabilidade.

- Adote uma estratégia de várias contas: isole ambientes para reduzir o raio de incidentes e simplificar as operações.
- Uso AWS Organizations: gerencie contas hierarquicamente, aplique políticas e habilite a governança centralizada.
- Planeje a escalabilidade: projete estruturas de contas refinadas e implemente medidas de redução de custos para o crescimento futuro.

Etapas de implementação

Um sistema de jogo implantado AWS deve usar várias contas que são organizadas logicamente para fornecer isolamento adequado, o que reduz o raio de ocorrência de problemas e simplifica as operações à medida que a infraestrutura do jogo aumenta. Contas da AWS essa infraestrutura de jogos hospedeiros geralmente é agrupada nos seguintes ambientes lógicos:

- Os ambientes de desenvolvimento de jogos são usados pelos desenvolvedores para desenvolver o software e os sistemas do jogo.
- Ambientes de teste ou garantia de qualidade (QA) são usados para realizar testes de integração, controle de qualidade manual e outros testes automatizados que devem ser conduzidos.
- Ambientes de preparação ou pré-produção são usados para hospedar o software completo para que os testes de carga e fumaça possam ser conduzidos antes do início da produção.
- Ambientes ao vivo ou de produção são usados para hospedar o software e a infraestrutura ativos e atender ao tráfego de produção dos players.
- Ambientes de serviços ou ferramentas compartilhados fornecem acesso a sistemas, softwares e ferramentas comuns que são usados por muitas equipes diferentes. Por exemplo, um repositório central de controle de origem auto-hospedado e um farm de criação de jogos podem estar hospedados em uma conta de serviços compartilhados.
- Ambientes de segurança são usados para consolidar registros centralizados e tecnologias de segurança que são usadas por equipes que se concentram na segurança na nuvem.

Para a infraestrutura de jogos ativada AWS, é recomendável criar contas separadas para cada ambiente de jogo (desenvolvimento, teste, preparação e produção), bem como contas para segurança, registro e serviços compartilhados centrais.

Normalmente, pequenos estúdios de desenvolvimento de jogos que gerenciam um número limitado de recursos de infraestrutura, geralmente algumas centenas de servidores ou menos, podem criar um Conta da AWS para cada um desses ambientes (por exemplo, uma conta de produção, uma conta de desenvolvimento e uma conta de teste). No entanto, à medida que a infraestrutura do jogo ou o tamanho da equipe aumentam com o tempo, esse modelo simplificado pode não ser bem dimensionado.

Ao configurar esses ambientes, considere que muitos AWS serviços compartilham Quotas de Serviço em nível de API e de recursos para uma conta inteira em uma determinada região. Isso deve ser considerado ao determinar como organizar logicamente as contas. Contas da AWS apenas incorrem em custos pelo consumo de serviços implantados neles. Portanto, isso fornece uma maneira de reduzir efetivamente a contenção de recursos e as cotas de serviços, principalmente à medida que seu jogo cresce e mais desenvolvedores precisam de acesso para criar e gerenciar recursos.

Com base em nossa experiência de trabalho com grandes estúdios de desenvolvimento de jogos que normalmente operam milhares de servidores com centenas de desenvolvedores acessando recursos, recomendamos que você crie uma estrutura de contas mais refinada, na qual os aplicativos individuais que dão suporte ao seu jogo tenham suas próprias contas de desenvolvimento, teste, preparação e produção. Como é difícil e demorado redesenhar sua estratégia de AWS várias contas após o lançamento do jogo devido à complexidade do planejamento e da migração de sistemas ativos, considere suas necessidades futuras de escalabilidade ao determinar a estrutura correta de várias contas.

Você pode usar AWS Organizationspara configurar uma hierarquia Contas da AWS, um agrupamento e definir unidades organizacionais (OUs) para aplicar políticas comuns de nível de UO a elas por meio de políticas de controle de serviço (). SCPs AWS Organizations gerencia e governa centralmente seu ambiente à medida que você cresce e escala seus recursos. Você pode criar novas contas e alocar recursos de forma programática, agrupar contas para organizar seus fluxos de trabalho, aplicar políticas a contas ou grupos para fins de governança e simplificar o faturamento usando um único método de pagamento para suas contas. Além disso, o Organizations é integrado a outros serviços para que você possa definir configurações centrais, mecanismos de segurança, requisitos de auditoria e compartilhamento de recursos entre contas em sua organização.

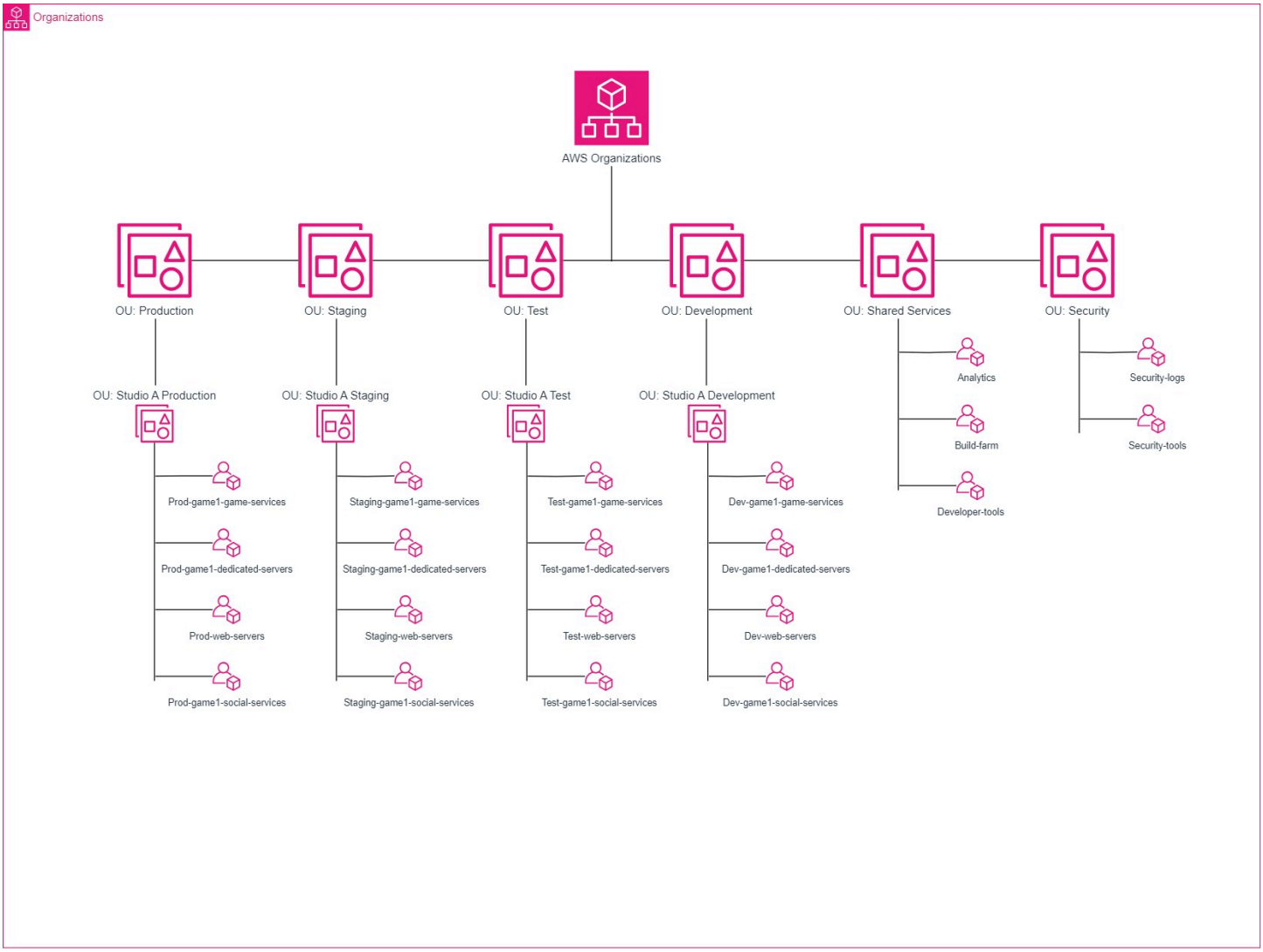
AWS Control Towerfornece uma maneira simples de configurar e controlar um ambiente seguro com várias contas, chamado de landing zone. A Control Tower cria sua landing zone usando AWS Organizations, trazendo gerenciamento e governança contínuos de contas, bem como as melhores práticas AWS de implementação com base na experiência de trabalhar com milhares de clientes à medida que eles migram para a nuvem. AWS Config, AWS Trusted Advisor, e AWS Security Hub CSPMsão serviços que fornecem uma visão agregada ou centralizada da higiene da sua conta.

Esse isolamento ajuda você a configurar permissões e proteções personalizadas ou individuais para cada ambiente de jogo. As contas de produção devem ter as proteções, as restrições de acesso, o monitoramento, os alertas e as ferramentas de segurança necessárias, enquanto as contas que não são de produção podem não exigir o mesmo nível de proteções e permissões. Ambientes de não produção podem ser automatizados para desligar recursos após o expediente e economizar custos. A separação de contas nesse nível de granularidade facilita o monitoramento dos custos de infraestrutura de cada um dos ambientes que suportam um jogo.

Veja a seguir um exemplo de uma estrutura de várias contas para uma empresa de jogos que usa AWS Organizations unidades organizacionais (OUs) para agrupar logicamente Contas da AWS em ambientes e estúdios separados. Neste exemplo, OUs são usados para agrupar contas com base em seu ambiente e, em seguida, com base no estúdio que opera o ambiente. Isso demonstra como você pode criar uma hierarquia aninhada para permitir que aplicativos e jogos separados sejam implantados em suas próprias contas dentro do ambiente (representado como OUs), o que pode ser útil se você desenvolver e operar vários jogos. Consulte a documentação e os whitepapers fornecidos na seção de recursos deste pilar para saber mais sobre estratégias adicionais que você pode considerar para organizar sua estratégia de várias contas.

Com base na discussão acima, o diagrama de exemplo abaixo pressupõe um estúdio de jogos (organização) que tenha um pipeline de desenvolvimento composto por 4 estágios (desenvolvimento, teste, preparação e produção). Para um determinado jogo (jogo1), cada um dos ambientes (OU) tem serviços individuais Contas da AWS para jogos, servidores de jogos dedicados, serviços sociais e servidores web. Os recursos executados em cada um Conta da AWS são relevantes para os respectivos subsistemas. Normalmente, cada jogo individual que usa esse tipo de pipeline de desenvolvimento replicaria essa ou uma estrutura similar. Contas da AWS

Além desses ambientes centrados em jogos OUs, há também a OU de serviços compartilhados e a OU de segurança. Eles OUs devem ser para toda a organização, não para cada jogo individual. Dessa forma, os jogos consumiriam os serviços compartilhados de ferramentas de desenvolvimento, dados e análises, como neste exemplo. Em seguida, envie os registros do aplicativo e do sistema para a Conta da AWS configuração de registros na OU de segurança.



Exemplo de estrutura de conta para ambientes de jogos

GAMEOPS02-BP02 Organize recursos de infraestrutura usando marcação de recursos

Para gerenciar e rastrear com eficácia seus recursos de infraestrutura AWS, use a marcação e o agrupamento adequados de recursos para identificar o proprietário, o projeto, o aplicativo, o centro de custos e outros dados de cada recurso. Os recursos marcados podem ser agrupados usando grupos de recursos, o que auxilia no suporte operacional.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Defina políticas de marcação. As estratégias típicas incluem tags de recursos para identificar o proprietário do recurso, como nome da equipe ou nome individual, nome do jogo, aplicativo ou projeto, nome do estúdio, ambiente (como desenvolvimento ou produção) e função do recurso (como servidor de banco de dados, servidor web, servidor de jogos dedicado, servidor de aplicativos ou servidor de cache). Você pode adicionar outras tags para ajudar nas necessidades comerciais e de TI. AWS Configtambém pode aplicar uma política de marcação no momento da criação e atualização do recurso. As tags e os grupos de recursos estão disponíveis nas operações Console de gerenciamento da AWS AWS CLI, the e API.

Etapas de implementação

- Marque recursos para identificar o proprietário, o projeto, o aplicativo, o centro de custos e outros dados relevantes.
- Implemente políticas de marcação, incluindo tags para proprietário, projeto, estúdio, ambiente e função do recurso.
- Use AWS Config para aplicar políticas de marcação e gerenciar tags por meio de Console de gerenciamento da AWS CLI e API.

Implantações de jogos

GAMEOPS03: Como você gerencia as implantações de jogos?

Gerencie implantações de jogos validando completamente os componentes reutilizados, conduzindo engenharia de desempenho regular e implementando testes de carga regulares em todo o ciclo de vida do desenvolvimento.

Práticas recomendadas

- GAMEOPS03-BP01 Valide e teste seus principais sistemas e infraestrutura de jogo existentes antes de reutilizá-los em seu jogo
- GAMEOPS03-BP02 Conduza a engenharia de desempenho antes de cada lançamento (ou pelo menos para lançamentos principais)
- GAMEOPS03-BP03 Teste de carga cedo e com frequência

- GAMEOPS03-BP04 Adote uma estratégia de implantação que minimize o impacto para os jogadores
- GAMEOPS03-BP05 Infraestrutura de pré-escala necessária para suportar os requisitos de pico

GAMEOPS03-BP01 Valide e teste seus principais sistemas e infraestrutura de jogo existentes antes de reutilizá-los em seu jogo

As organizações tendem a reutilizar componentes existentes e o código-fonte de jogos anteriores para economizar tempo e custos de desenvolvimento. Esses componentes e códigos legados podem não estar sujeitos a uma revisão completa ou ter testes de integração detalhados e, em vez disso, confiar em seu desempenho anterior.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Embora a reutilização ajude a melhorar a produtividade, ela também pode introduzir o risco de reintroduzir problemas anteriores de desempenho e estabilidade em um novo projeto. Portanto, ao reutilizar componentes existentes e o código-fonte de jogos anteriores, testes robustos devem ser implementados.

Etapas de implementação

- Identifique o código e os componentes reutilizados: Catalogue o código-fonte, as bibliotecas e os componentes que estão sendo reutilizados de jogos anteriores. Faça uma distinção clara entre código mantido ativamente e código obsoleto
- Documente o comportamento original e os problemas conhecidos: registre as características de desempenho originais, as limitações funcionais e os bugs conhecidos ou incidentes de produção associados aos componentes reutilizados.
- Faça uma revisão completa do código: faça uma análise técnica detalhada dos componentes reutilizados, especialmente aqueles que tiveram problemas no passado ou estão mal documentados.
- Substitua ou refatore componentes legados de alto risco: priorize a substituição ou atualização de componentes legados que tenham um histórico de problemas ou não possam mais ser mantidos, em vez de confiar em soluções alternativas na produção.

- Realize testes de integração e compatibilidade: valide os componentes reutilizados dentro do contexto dos sistemas do novo jogo. Verifique se eles interagem adequadamente com novos módulos, ferramentas APIs e.

GAMEOPS03-BP02 Conduza a engenharia de desempenho antes de cada lançamento (ou pelo menos para lançamentos principais)

A engenharia de desempenho é o processo de monitorar várias métricas operacionais importantes de um aplicativo para descobrir oportunidades de otimização que podem melhorar ainda mais o desempenho do aplicativo. Esse é um processo iterativo que começa com o teste, seguido pela otimização do código, suas dependências, processos associados, seu sistema operacional hospedeiro e a infraestrutura subjacente.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Para realizar uma análise mais profunda do desempenho do aplicativo, integre uma ferramenta de monitoramento de desempenho do aplicativo (APM) ou de depuração no código do aplicativo que possa isolar problemas e reduzir o tempo de solução de problemas rastreando seu comportamento em busca de anomalias nos fluxos do aplicativo. As ferramentas de APM também são capazes de identificar métodos de desempenho lento e operações externas.

AWS X-Rayauxilia os desenvolvedores em suas atividades de engenharia de desempenho, como identificar gargalos de desempenho e analisar e depurar erros de produção. Você pode usar o X-Ray para entender o desempenho do aplicativo e dos serviços subjacentes e identificar e solucionar a causa raiz dos problemas e erros de desempenho. Por meio de várias rodadas de testes de carga, nas quais o aplicativo e sua infraestrutura são gradualmente carregados com tráfego sintético de jogadores, vários gargalos do sistema, erros de aplicativos, exceções, problemas de sistema operacional e outros problemas são identificados que podem não ter sido encontrados durante outros testes de controle de qualidade.

Para eventos críticos, como lançamentos de jogos, lançamentos de conteúdo, promoções e grandes eventos no jogo, use o AWS Countdown, que fornece orientação de implementação com base em manuais criados por especialistas em jogos para verificar a prontidão operacional, mitigar riscos potenciais e planejar as necessidades de capacidade.AWS O Countdown também tem uma opção de suporte premium que oferece suporte aprimorado e opções como engenheiros para otimizar sua infraestrutura.

Etapas de implementação

- A engenharia de desempenho envolve avaliar e monitorar as principais métricas operacionais para verificar se o código, os processos, o sistema operacional e a infraestrutura do seu aplicativo estão funcionando conforme o esperado. A análise de pré-produção também ajuda a definir o desempenho básico em diferentes níveis de uso simulado.
- Descubra e acompanhe as principais métricas, como utilização, serviços, E/S, processos e outras, usando ferramentas do sistema, como sar, top, vmstat, sysstat, netstat e Performance Monitor.
- Monitore o desempenho e o comportamento do seu aplicativo usando ferramentas de APM AWS X-Ray para isolar problemas, identificar gargalos e depurar erros de produção.
- Para eventos críticos, como lançamentos de jogos, assine o AWS Countdown (IEM) para obter orientação arquitetônica e operacional, suporte operacional sob demanda e identificar riscos e planejar mitigações.

GAMEOPS03-BP03 Teste de carga cedo e com frequência

O teste de carga é o processo de simular o tráfego do mundo real em um sistema para avaliar sua confiabilidade e desempenho.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

O teste de carga é um fator-chave para desenvolver uma linha de base de desempenho para seus recursos e entender a capacidade do seu sistema, que pode orientar a previsão financeira, o projeto de arquitetura, a alocação de recursos, as configurações de escalabilidade automatizadas e as atividades de pré-escalonamento pós-lançamento. Os benefícios adicionais incluem:

- Infraestrutura otimizada: os recursos podem estar superprovisionados ou subprovisionados. Compreender os recursos necessários resultará em custos mais baixos e menos infraestrutura para gerenciar.
- Prontidão para escalabilidade: certos mecanismos e recursos podem levar os usuários a um jogo rapidamente. Saber quando e como escalar pode ser a diferença entre atender adequadamente ao aumento da demanda e perder jogadores. Use os resultados do teste de carga para preparar runbooks com limites do sistema, pontos de alerta e pontos de alerta críticos em diferentes níveis de escala.

- Código de maior qualidade: problemas como interferência excessiva entre serviços, chamadas de banco de dados sem lote, algoritmos ineficientes, vazamentos de memória e problemas de degradação do serviço às vezes são mais simples de identificar em grande escala.
- Validação de comportamento: injetar diferentes tipos de falhas em seus testes pode validar o comportamento esperado do sistema ou descobrir problemas de tratamento de erros que precisam ser corrigidos.

Idealmente, os desenvolvedores devem realizar testes de carga em vários pontos do processo de desenvolvimento, pois cada um pode gerar benefícios diferentes: desde o início, eles orientam as decisões de arquitetura e os esforços de refatoração, enquanto é mais barato e fácil fazer alterações. Ao final de cada sprint ou iteração, eles validam o desempenho do aplicativo com os recursos e funcionalidades mais recentes.

Antes da implantação na produção, testes de carga em grande escala simulando padrões de uso esperados no mundo real confirmam a capacidade do sistema de lidar com a carga de trabalho de produção. Após a implantação, testes de carga periódicos monitoram o desempenho do sistema e identificam alterações ou gargalos que possam surgir ao longo do tempo.

Para simular o tráfego de jogadores, você precisa de clientes ou bots leves que emulem os fluxos do cliente do jogo e façam transações com o back-end do jogo para simular o comportamento do jogador no mundo real. Esses dados geralmente são capturados por meio de registros de jogo e dados gerados por testes de controle de qualidade conduzidos por humanos, bem como por meio de testes alfa ou beta de escala limitada do mundo real, nos quais jogadores reais são convidados a jogar uma versão do jogo com acesso antecipado.

É importante registrar o comportamento do sistema em um runbook operacional para auxiliar na solução de possíveis falhas no futuro e reter métricas de desempenho com as quais futuros testes de carga possam ser comparados. Também é recomendável que uma equipe humana de controle de qualidade teste o jogo durante o teste de carga, pois eles podem descobrir problemas que os bots não conseguem identificar e que as métricas não refletem.

AWS O Fault Injection Service é um serviço totalmente gerenciado para executar experimentos de injeção de falhas que facilitam a melhoria do desempenho, da observabilidade e da resiliência de um aplicativo. Os experimentos de injeção de falhas são usados na engenharia do caos, que é a prática de sobrecarregar uma aplicação em ambientes de teste ou produção criando eventos disruptivos, como aumento repentino no consumo de CPU ou memória, observar como o sistema responde e implementar melhorias. Os experimentos de injeção de falhas ajudam as equipes a criar

as condições reais necessárias para descobrir bugs ocultos, monitorar pontos cegos e gargalos de desempenho que são difíceis de encontrar em sistemas distribuídos.

Etapas de implementação

- Configure um ambiente de teste de carga distribuído usando o Guidance for Kubernetes-Bases Game Load Testing.
- Personalize e implante os pods de controle e de trabalho do Locust no cluster EKS usando os arquivos de implantação fornecidos, permitindo a geração de carga escalável e gerenciável.
- Registre o comportamento e as métricas do sistema durante o teste de carga em um runbook operacional para ajudar na futura solução de problemas e estabelecer linhas de base de desempenho.
- Use experimentos de injeção de falhas para simular interrupções no mundo real e descobrir problemas ocultos no desempenho, observabilidade e resiliência do sistema.

GAMEOPS03-BP04 Adote uma estratégia de implantação que minimize o impacto para os jogadores

Incorpore uma estratégia de implantação para o software e a infraestrutura do seu jogo que minimize a quantidade de tempo de inatividade que mantém os jogadores fora do seu jogo. Embora certos tipos de atualizações possam exigir a instalação de novas atualizações no cliente do jogo, desenvolva o jogo para minimizar ou evitar a necessidade de tempo de inatividade durante as implantações.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Uma das etapas mais importantes a serem consideradas ao desenvolver uma estratégia de implantação de jogos é determinar como sua infraestrutura de jogos será gerenciada. Gerencie sua infraestrutura de jogos usando uma ferramenta de infraestrutura como código (IaC), como AWS CloudFormationo Terraform da Hashicorp, para reduzir os erros humanos durante a preparação do ambiente. Os modelos de infraestrutura podem ser implantados e testados em pipelines automatizados, o que cria consistência na configuração de diferentes ambientes de jogo.

Há várias estratégias de implantação que podem ser usadas em um jogo:

Substituição contínua

O objetivo principal de uma substituição contínua para implantação é realizar o lançamento sem encerrar o jogo e sem afetar os jogadores. É importante que a atualização ou as alterações a serem executadas sejam compatíveis com versões anteriores e funcionem de forma adjacente às versões anteriores do sistema.

Nessa implantação, as instâncias do servidor são substituídas incrementalmente (substituídas ou implantadas) por instâncias que executam a versão atualizada. Essa substituição contínua pode ser realizada de algumas maneiras diferentes. Por exemplo, para implementar atualizações contínuas em uma frota de servidores de jogos dedicados, uma abordagem típica envolve criar um novo grupo de EC2 instâncias do Auto Scaling que contém a nova versão de compilação do servidor de jogos implantada nelas e, em seguida, direcionar gradualmente os jogadores para as sessões de jogo hospedadas nessa nova frota de servidores. Se houver uma atualização do cliente de jogo associada que seja necessária como pré-requisito para usar a nova versão do servidor de jogo, você deverá incluir uma verificação de validação para verificar se somente os jogadores que têm essa nova atualização do cliente de jogo instalada são encaminhados para essas sessões de jogo.

As frotas de servidores (por exemplo, grupos de EC2 Auto Scaling) que contêm a versão antiga de compilação do servidor de jogos só são removidas do serviço depois de serem esgotadas das sessões ativas dos jogadores de maneira elegante, normalmente configurando métricas de servidor individualizadas que permitem que as equipes de operações do jogo automatizem esse processo. Como alternativa, para reduzir a quantidade de infraestrutura e o tempo para realizar uma implantação contínua, uma abordagem alternativa pode ser realizada em que as instâncias de produção existentes sejam removidas do serviço, atualizadas com a nova construção do servidor de jogos e depois colocadas de volta na frota de produção. Essa abordagem reduz a quantidade de infraestrutura necessária, mas também aumenta o risco, pois o número de servidores de jogos ao vivo disponíveis para os jogadores é reduzido à medida que os servidores são substituídos.

Esse modelo também pode ser usado para realizar implantações contínuas em serviços de back-end, como bancos de dados, caches e servidores de aplicativos que não hospedam jogos. Desde que esses serviços sejam implantados de maneira altamente disponível com várias instâncias em cluster, a complexidade das implantações nesses serviços deve ser menor do que as implantações em servidores de jogos dedicados.

Implantação azul/verde

O objetivo principal de uma blue/green implantação em um jogo é minimizar o tempo de inatividade e, ao mesmo tempo, permitir a reversão segura da implantação anterior, caso sejam identificados problemas. É adequado para implantações em que duas versões do back-end do jogo são compatíveis e podem atender aos jogadores simultaneamente.

Na estratégia blue/green de implantação, dois ambientes idênticos (azul e verde) são configurados. A versão existente do jogo é rotulada como azul, enquanto a nova versão do jogo que é o alvo de implantação é rotulada como verde. Quando o ambiente verde estiver pronto para a migração, você poderá configurar sua camada de roteamento para reverter o tráfego para o ambiente verde, mantendo o ambiente antigo (azul) disponível caso seja necessário um failback. Nesse cenário, as atualizações de roteamento podem exigir a atualização do serviço de matchmaking para configurá-lo para começar a enviar sessões de jogo para a nova frota ou, no caso de serviços de back-end de jogos, isso pode ser atualizar os registros DNS no Amazon Route 53 para seu serviço ou mudar os pesos do balanceador de carga do aplicativo para enviar tráfego para seu novo grupo-alvo.

Uma das desvantagens da estratégia de blue/green implantação é o custo inerente do ambiente de espera devido à infraestrutura adicional necessária durante a execução da implantação. Uma opção para mitigar esse custo adicional de infraestrutura é considerar a adoção de uma variante de blue/green implantação em que o novo software de jogo é implantado nos mesmos servidores que já estão implantados na produção. Nesse cenário, um novo processo de servidor verde pode ser iniciado com o novo software junto com o processo de servidor azul existente, com a transição ocorrendo entre os processos do servidor e não entre uma infraestrutura física separada. Essa abordagem também pode acelerar as implantações de jogos em uma grande quantidade de infraestrutura, eliminando a necessidade de esperar que novos servidores sejam lançados na nuvem. Para obter as melhores práticas sobre essa abordagem de implantação, consulte Implantações azul/verde ativadas. AWS

Implantação do Canary

A implantação do Canary é útil para desenvolvedores de jogos, pois a estratégia pode ser aplicada para lançar uma versão alfa ou beta inicial de um jogo, ou um recurso do jogo, como um novo modo de jogo, mapa ou desafio, para um conjunto restrito ou pequeno de jogadores em produção. Essa implantação é chamada de canário. O lançamento pode ter rastreamento e relatórios adicionais, portanto, quando jogadores reais jogam esse jogo ou recurso, sua telemetria de jogo é coletada e analisada em busca de anomalias e problemas.

Para novos recursos, os jogadores não são notificados consistentemente sobre isso, e a telemetria do jogo é a principal fonte usada para determinar se os jogadores estão enfrentando problemas e se o lançamento deve ser revertido. Ao mesmo tempo, se nenhum problema significativo for identificado, o recurso poderá ser implementado para mais jogadores para obter dados adicionais. Se os jogadores forem notificados, eles poderão ser solicitados a fornecer feedback regular sobre sua experiência. Idealmente, essa atividade de teste seria coordenada por uma equipe de operações ao vivo.

Como estratégia, o canary deployment também pode ser usado em lançamentos padrão para disponibilizar gradualmente um novo recurso para os jogadores. Uma vantagem potencial em relação ao blue/green ambiente padrão é que não é necessário um segundo ambiente em grande escala. A capacidade do novo ambiente reduzido determina quantos jogadores devem ser integrados ao novo recurso. Antes de adicionar mais jogadores, a capacidade deve ser dimensionada adequadamente. Mesmo que se espere que essa blue/green técnica personalizada custe comparativamente menos do que o azul/verde padrão, estima-se que ela tenha um custo que pode ser maior do que a técnica de substituição contínua de implantações de canários.

Execute apenas um único canário em um ambiente de produção e concentre-se em seus dados e feedback. Se vários canários forem implantados, isso complica a solução de problemas e o isolamento de problemas na produção e prejudica a qualidade dos conjuntos de dados e do feedback que estão sendo coletados.

Uma variação do canário ocorre quando um ou mais experimentos (geralmente testes de interface do usuário) são executados por meio de implantações específicas, em que um conjunto de servidores de back-end do jogo serve uma versão de um recurso e outro conjunto do mesmo tamanho fornece outra versão do mesmo recurso. Nenhuma infraestrutura adicional ou especial é criada para isso, e somente os pacotes escolhidos de servidores de back-end recebem essas atualizações. O resultado dos experimentos é observar como os jogadores reagem a cada uma das versões do mesmo recurso, determinar se há um consenso geral de gostar ou não gostar e observar se há problemas identificados com sua usabilidade ou funcionalidade. Esses experimentos estratégicos também são chamados de A/B testes, e o processo geral é chamado de teste A/B. Após a conclusão desses experimentos, os dados de teste necessários são coletados antes de reverter para a versão atual do sistema de back-end do jogo nos servidores usados para os testes.

Implantações tradicionais antigas

No estilo tradicional de implantação, durante uma janela de manutenção programada, o jogo é encerrado e os jogadores conectados são eliminados ou esgotados antes que as instâncias do servidor no back-end do jogo sejam atualizadas com as versões de código mais recentes. Essa implantação afeta os jogadores toda vez que é realizada, e os jogadores devem ser notificados antes do cronograma. Como resultado, esse modelo causa o maior impacto ao jogador e deve ser evitado sempre que possível.

Depois que a atualização do jogo for implantada, o jogo poderá ser testado antes de abri-lo para os jogadores, que estariam esperando a reabertura do jogo. Isso pode causar um aumento no tráfego quando os jogadores tentam fazer login e jogar em um curto período de tempo. Portanto, se o jogo

não foi projetado para lidar com esses picos de tráfego, você pode optar por permitir gradualmente que os jogadores voltem ao jogo em lotes.

Como alternativa, você pode optar por provisionar demais a infraestrutura para sustentar o pico inicial de tráfego e, depois que o tráfego do jogo diminuir, os recursos poderão ser reduzidos. Se necessário, realize esse tipo de implantação fora do horário de pico, quando o número de jogadores é menor. A manutenção programada com frequência, bem como a manutenção prolongada, inerentemente acarreta o risco de desgaste do jogador e de possível perda de receita. Os jogadores também esperam mudanças após um novo lançamento e podem perder a confiança no jogo ao retornar após um período de inatividade.

Etapas de implementação

- Minimize o tempo de inatividade: implemente estratégias de implantação que reduzam o tempo de inatividade e mantenham os jogadores no jogo.
- Infraestrutura como código (IaC): use ferramentas como AWS CloudFormation o Terraform para gerenciar a infraestrutura do jogo e reduzir os erros humanos.
- Estratégias de implantação: use uma ou uma combinação de substituições contínuas, azul/verde e canário para fornecer atualizações suaves e reduzir o impacto do jogador.

GAMEOPS03-BP05 Infraestrutura de pré-escala necessária para suportar os requisitos de pico

Expanda a infraestrutura antes dos eventos de jogos em grande escala para garantir que você possa lidar com o aumento repentino na demanda dos jogadores.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Além dos lançamentos de novos jogos, os jogos ao vivo geralmente oferecem eventos, promoções, novos conteúdos e lançamentos de temporada no jogo como exemplos de formas de manter e melhorar o engajamento dos jogadores. Essas atividades geram um alto volume de tráfego de jogadores durante o evento ou promoção. A empresa espera atingir ou superar as metas pretendidas para o evento, e a infraestrutura do jogo deve sustentá-la e apoiá-la durante esse período.

Prepare sua infraestrutura com antecedência para poder suportar a carga prevista de jogadores que você experimentará durante eventos de grande escala. Para se preparar, as equipes de operações

de jogos devem se coordenar com as partes interessadas em vendas e marketing para estimar a demanda projetada que será gerada em um próximo evento, analisando a simultaneidade de jogadores anteriores, as métricas de engajamento e os dados de vendas. Se o evento for para o lançamento de um novo jogo, as equipes de operações do jogo devem trabalhar com essas partes interessadas para identificar projeções realistas da escala que elas esperam. Embora possa ser difícil prever o sucesso de um jogo, é importante que todos entendam quais são as expectativas de sucesso para que a infraestrutura possa ser escalada e testada para dar suporte a essas metas.

Muitos jogos optam por lançar em etapas, começando com um lançamento suave, abrindo o jogo para um pequeno número de jogadores e, em seguida, escalando organicamente os jogadores em cada estágio, antes de um lançamento público completo. Durante o período de lançamento temporário, monitore, identifique, acompanhe e resolva problemas enquanto refina suas projeções para o lançamento público.

Para estimar adequadamente os requisitos de infraestrutura, colete dados por meio de testes de carga e desempenho executados nos back-ends do seu jogo em execução na produção ou em um ambiente de teste semelhante ao de produção antes do lançamento do jogo. Várias rodadas desses testes devem ser executadas para simular diferentes condições do jogo e validar se o back-end pode suportar a carga na maioria das condições.

Para conseguir isso, os desenvolvedores podem criar bots de jogabilidade que percorrem vários fluxos de trabalho no jogo e emulam diferentes condições. Esses testes devem inspecionar as diferentes camadas do sistema do back-end do jogo para que cada camada e componente seja testado e os detalhes sejam registrados. Use os dados coletados desses testes para provisionar o plano para o lançamento do jogo.

Os pontos únicos de falha (SPOF) devem ser identificados e removidos sempre que possível, tornando o aplicativo altamente disponível e tolerante a falhas. Use testes de carga para identificar, SPOFs emulando falhas em diferentes camadas ascendentes e posteriores e verificando o comportamento do jogo e de outros componentes.

Além da infraestrutura estimada necessária a ser provisionada para o lançamento do jogo, o evento no jogo ou os preparativos da promoção, configure o sistema para escalar automaticamente sob demanda. Defina, configure e monitore os limites de eventos de escalabilidade para permitir que o back-end do jogo seja dimensionado para sustentar um alto volume de tráfego de jogadores. Para tráfego variável, o pré-provisionamento é melhor porque pode não haver tempo suficiente para a expansão horizontal. O dimensionamento manual pode ser necessário durante os lançamentos iniciais do jogo, o que gera uma demanda maior do que a prevista, mais rápido do que sistemas automatizados podem escalar recursos.

Ativado AWS, as organizações devem solicitar Quotas de Serviço mais altas para os serviços que usam no back-end do jogo. As Cotas de Serviço são configuradas para contas para evitar que os clientes instalem ou escalem inadvertidamente mais infraestrutura do que o pretendido. Quando um jogo em execução em uma conta atinge o limite superior da cota de serviço configurada nessa região, o serviço limita as solicitações além da cota provisionada e interrompe as provisões. Os aceleradores podem causar erros não intencionais ou inesperados e prejudicar a experiência do jogador. Monitore, acompanhe e revise regularmente os limites da cota de serviço dos serviços usados pelo jogo em produção para evitar limitações. Quando o uso ultrapassa um limite tolerável de cota de serviço, um aumento na cota pode ser solicitado gerando um caso de suporte na Central de Suporte do Console, depois de fazer login na conta afetada ou usando a API de suporte.

Para eventos críticos, como lançamentos de jogos, lançamentos de conteúdo, promoções e grandes eventos do jogo, use o AWS Countdown. O Countdown fornece orientação de implementação com base em manuais criados por especialistas em jogos para fornecer prontidão operacional, mitigar riscos potenciais e planejar as necessidades de capacidade.AWS O Countdown também tem uma opção de suporte premium que oferece suporte aprimorado e opções como engenheiros para otimizar sua infraestrutura.

Se você estiver lançando um jogo hospedado na Amazon GameLift, revise as listas de verificação de pré-lançamento para se preparar.

Etapas de implementação

- Expanda a infraestrutura com antecedência: prepare a infraestrutura com antecedência para eventos de jogos em grande escala para lidar com aumentos repentinos na demanda dos jogadores.
- Estime a demanda: coordene as vendas e o marketing para estimar a demanda projetada usando dados anteriores de jogadores e projeções realistas.
- Teste de carga e remoção de SPOF: realize várias rodadas de testes de carga para validar a capacidade de back-end, identificar pontos únicos de falha e configurar adequadamente o escalonamento automatizado.

Monitoramento de saúde

GAMEOPS04: Como você monitora a saúde do jogo?

Monitore a integridade do jogo implementando uma instrumentação abrangente para detectar e rastrear problemas que afetam os jogadores, incluindo registro de atividades do lado do cliente, monitoramento de serviços de back-end e relatórios de erros. Use uma combinação de AWS ferramentas como a Amazon CloudWatch e AWS X-Ray, além de soluções de terceiros, para ajudar a identificar e resolver problemas rapidamente.

Práticas recomendadas

- GAMESOPS04-BP01 Instrumenta o jogo para detectar e monitorar problemas que afetam os jogadores

GAMESOPS04-BP01 Instrumenta o jogo para detectar e monitorar problemas que afetam os jogadores

Além de responder às redes sociais e aos relatos de problemas dos jogadores, instrumente seu jogo com soluções de monitoramento para detectar e investigar problemas que afetam os jogadores.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Nenhuma quantidade de testes pode identificar todos os problemas em um jogo. Os jogos geralmente são lançados com problemas conhecidos que devem ser corrigidos gradualmente na próxima versão do jogo. Problemas conhecidos e reproduzíveis são fáceis de resolver e corrigir. Para ajudar a identificar esses problemas, os clientes do jogo devem implementar o rastreamento de atividades do jogador, o registro de aplicativos e os relatórios em vários locais estratégicos para ajudar a equipe de back-end a identificar problemas do lado do cliente. A capacidade de encontrar esses problemas com antecedência ajuda os desenvolvedores de jogos a solucionar e corrigir o problema antes que ele se espalhe. Os dados e registros relatados pelo código de rastreamento nunca devem incluir informações de identificação pessoal (PII) e devem conter apenas metadados específicos do jogo que auxiliem na depuração.

Implemente uma solução de observabilidade para detectar e responder a problemas como falhas no jogo ou bugs. Você pode usar o Amazon CloudWatch Synthetics para criar canários que possam monitorar a integridade de seus serviços de back-end de jogos voltados para jogadores. Você pode instrumentar seus serviços de back-end AWS X-Raypara rastrear solicitações em serviços distribuídos e enviar seus registros e métricas personalizados para a Amazon CloudWatch.

Soluções de terceiros, como Backtrace.io e Sentry, são soluções populares para relatórios de erros em jogos. As soluções de monitoramento de desempenho de aplicativos (APM) de parceiros como New Relic, Splunk, Datadog e Honeycomb.io também são populares.

A equipe de operações ao vivo e os gerentes da comunidade do jogo também devem monitorar várias redes sociais e canais para verificar o feedback, as reclamações e os relatórios de bugs dos jogadores, além dos canais oficiais de suporte. Analise e tente reproduzir cada reclamação específica do jogo ou envie-a para a equipe de controle de qualidade para análise. Se for reproduzível, encaminhe o problema aos desenvolvedores do jogo para que solucionem e corrijam antes que ele afete a maior base de jogadores.

Etapas de implementação

- Implemente soluções de monitoramento: use ferramentas de monitoramento para detectar problemas que afetam os jogadores e responder rapidamente.
- Rastreie a atividade e os registros dos jogadores: instrumente os clientes do jogo para registrar a atividade do jogador e relatar problemas, além de verificar se nenhuma informação de identificação pessoal (PII) está incluída.
- Use AWS ferramentas e ferramentas de terceiros: use ferramentas como CloudWatch X-Ray e soluções de terceiros para relatórios de erros e monitoramento de desempenho, e monitore as mídias sociais em busca de feedback de jogadores e relatórios de bugs.

Testes de carga

GAMEOPS05: O que você deve considerar ao testar a carga de um jogo?

Ao testar a carga de um jogo, considere o estágio de teste, a arquitetura de geração de carga e a estrutura de teste apropriados para avaliar com eficácia o desempenho e a escalabilidade do sistema. Escolha a combinação certa de tempo (desenvolvimento inicial, sprints, pré-produção ou pós-implantação), infraestrutura (EKSEC2, Fargate ou Lambda) e ferramenta de teste (JMeterLocust, Grafana K6 ou Gatling) para se alinhar às características exclusivas e às metas de desenvolvimento do seu jogo.

Práticas recomendadas

GAMEOPS05-BP01 Escolha o estágio, a arquitetura e a estrutura de teste de carga certos para atingir seus objetivos

A abordagem para testar a carga de um jogo pode variar significativamente dependendo de muitos fatores, incluindo o estágio do processo de desenvolvimento em que ele é executado, a arquitetura do próprio sistema gerador de carga e a escolha da estrutura de teste de carga. O momento em que ela é conduzida, seja nas fases iniciais, durante os sprints iterativos, antes da implantação na produção ou após a implantação, moldará as metas e o foco dos esforços de teste. Diferentes projetos de infraestrutura de geração de carga têm seus próprios prós e contras, e a seleção da estrutura de teste de carga influencia muito os recursos, a facilidade de uso e as integrações disponíveis para o processo de teste. Ao alinhar cuidadosamente esses elementos, as equipes de desenvolvimento podem adaptar a abordagem de teste de carga às características exclusivas do jogo, extrair os insights de desempenho mais valiosos e proporcionar uma experiência tranquila para seus jogadores.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Teste de carga em diferentes estágios de desenvolvimento

A realização de testes exploratórios de carga no início das fases de desenvolvimento pode validar a arquitetura subjacente do sistema. Isso ajuda os desenvolvedores a tomarem decisões informadas sobre a infraestrutura, o design do banco de dados e a topologia da rede do jogo antes que um extenso trabalho de implementação seja concluído. Os testes de carga identificam riscos e criam uma linha de base de desempenho, potencialmente minimizando a necessidade de retrabalho dispendioso e dívidas técnicas mais tarde no ciclo de vida do desenvolvimento. Eles também podem promover uma compreensão compartilhada dos requisitos de desempenho do jogo entre a equipe, levando a uma melhor colaboração e tomada de decisões. Em última análise, o teste de carga durante as fases iniciais cria uma base sólida para um jogo de alto desempenho, escalável e resiliente, ajudando a aprimorar a experiência geral do jogador.

No final de cada sprint ou iteração, o teste de carga pode avaliar o impacto no desempenho dos novos recursos, correções de bugs e outras mudanças introduzidas no ciclo mais recente. Essa abordagem direcionada permite que as equipes de desenvolvimento identifiquem rapidamente

as regressões ou degradações de desempenho introduzidas pelas atualizações mais recentes, permitindo que resolvam esses problemas antes que eles se propaguem mais adiante e mantendo um nível consistente de qualidade e desempenho.

Antes da implantação na produção, testes de carga robustos ajudam as equipes a validar a capacidade do sistema de lidar com as condições de tráfego e carga previstas no mundo real. Eles podem descobrir gargalos de escalabilidade ou restrições de recursos na infraestrutura de produção e oferecer a oportunidade de otimizar o desempenho do jogo, criando uma experiência de usuário suave e responsiva desde o primeiro dia. Os insights obtidos com os testes de carga de pré-lançamento podem mitigar os riscos do dia do lançamento e informar o planejamento contínuo da capacidade, que estabelece a base para a sustentabilidade e a escalabilidade de longo prazo do jogo.

O teste de carga de um jogo que já está em produção permite que as equipes monitorem o desempenho do jogo e identifiquem regressões ou degradações de desempenho que possam ocorrer ao longo do tempo. Isso permite que eles resolvam os problemas de forma proativa antes que eles afetem a experiência do jogador e afetem negativamente a retenção de usuários. Além disso, o teste de carga na produção valida a eficácia dos esforços de otimização de desempenho ou do dimensionamento da infraestrutura que foi implementado. Esse processo fornece uma experiência de jogo de alta qualidade, responsiva e escalável para os jogadores, mesmo à medida que o jogo evolui e amadurece.

Arquiteturas geradoras de carga

O design da arquitetura de geração de carga para testes de carga de jogos pode assumir várias formas, cada uma com seu próprio conjunto de vantagens e considerações.

No nível mais básico, as EC2 instâncias autogerenciadas da Amazon podem ser provisionadas e configuradas para atuar como geradores de carga. Com uma abordagem de nós de controle e nós de trabalho, você pode configurar várias instâncias geradoras de carga, cada uma executando seu próprio script de teste e gerenciada em geral por uma única instância de controle. A arquitetura pode ser ampliada e gerar mais carga sem aumentar a complexidade criando nós de trabalho adicionais, mas essa abordagem prática exige que as equipes lidem com o provisionamento, a configuração e o gerenciamento da infraestrutura subjacente.

Para uma abordagem mais escalável e orquestrada, você pode usar clusters Kubernetes do Amazon EKS para gerenciar e distribuir a carga de trabalho de teste de carga em uma frota de agentes de carga baseados em contêineres. Os recursos de escalabilidade automática do Kubernetes podem

ser usados para lidar com o escalonamento dos pods geradores de carga, enquanto as próprias equipes configuram e gerenciam as instâncias subjacentes no cluster que hospeda os pods. EC2

Como alternativa, a natureza sem servidor do AWS Fargatepode acelerar e simplificar a configuração do teste de carga, abstraindo o gerenciamento da infraestrutura e, ao mesmo tempo, fornecendo a escalabilidade e a flexibilidade necessárias. Para soluções híbridas em que já existe um cluster Kubernetes local e gerador de carga, mas pode ser necessária capacidade adicional, o EKS Anywhere pode gerenciar os dois clusters como um só. Console de gerenciamento da AWS

Você também pode usar AWS Lambdafunções de acordo com seus requisitos e metas. As funções Lambda são relativamente fáceis de configurar e escalar sem a necessidade de provisionar e gerenciar recursos adicionais. Eles também permitem a criação de cenários de teste mais complexos e dinâmicos devido à profunda integração com outros AWS serviços. No entanto, as funções Lambda têm limites nas funções simultâneas e no tempo de execução (15 minutos), o que pode restringir a escala e a duração do teste de carga que pode ser realizado. As latências de inicialização a frio também podem afetar a precisão dos resultados, e as limitações de recursos do Lambda podem não ser adequadas para cargas de trabalho de teste de carga altamente exigentes.

Os estúdios que desejam usar uma solução pré-construída podem usar o Distributed Load Testing on. AWS Essa solução usa o Amazon ECS on AWS Fargate para implantar contêineres que podem executar simulações de dezenas de milhares de usuários conectados. Você pode usar isso para iniciar rapidamente sua infraestrutura de teste de carga no estilo IAC usando AWS CloudFormation.

Estruturas de teste de carga

Não há duas estruturas de teste de carga construídas da mesma forma. Alguns têm interfaces gráficas intuitivas para criação de testes, enquanto outros são totalmente baseados em linha de comando. Uma ferramenta pode ser flexível e eficiente, mas exigir tempo e esforço para ser configurada e gerenciada, e outra pode ser sem servidor, mas limitada nos testes que pode criar e executar. Alguns gostam de grandes comunidades e muitos tutoriais, embora não tenham sido comprovados em campo, contrastando fortemente com outros que podem ser testados em produção, mas carecem de suporte ou documentação da comunidade. Escolha a estrutura que atinja o equilíbrio certo para você e sua equipe. Algumas opções populares são:

- Apache JMeter: estrutura popular de teste de carga de código aberto baseada em Java devido ao seu conjunto robusto de recursos e facilidade de uso. Sua capacidade de simular cenários complexos de usuário, uma ampla variedade de protocolos suportados, relatórios abrangentes e histórico comprovado são JMeter uma escolha confiável para testes de carga.

- Locust: estrutura de teste de carga moderna e distribuída construída em uma arquitetura orientada a eventos, tornando-a eficiente e eficiente em termos de recursos. Os testes são escritos em Python, permitindo cenários de teste flexíveis que aproveitam milhares de poderosas bibliotecas de terceiros, enquanto permanecem amigáveis e simples de ler.
- Grafana K6: poderosa estrutura de teste de carga que combina facilidade de uso com recursos avançados. Seu suporte para geração de carga distribuída, scripts flexíveis e integração perfeita com o Grafana para visualização de dados tornam o Grafana K6 uma escolha atraente.
- Gatling: estrutura de teste de carga de código aberto conhecida por seu desempenho e escalabilidade. Sua linguagem específica de domínio (DSL) baseada em Scala permite que os desenvolvedores criem scripts de teste de carga concisos e fáceis de manter, e seus robustos recursos de geração de relatórios e análises fornecem informações detalhadas do sistema em teste.

Etapas de implementação

- Estágios de teste de carga: realize testes de carga em vários estágios de desenvolvimento (desenvolvimento inicial, sprints, pré-produção e pós-implantação) para validar o desempenho do sistema e identificar problemas.
- Arquiteturas de geração de carga: escolha as arquiteturas de geração de carga apropriadas (EKSEC2, Fargate ou Lambda) com base nas necessidades de escalabilidade, nas preferências de gerenciamento e nos requisitos de teste específicos.
- Estruturas de teste de carga: selecione uma estrutura de teste de carga (como Locust JMeter, Grafana K6 ou Gatling) que equilibre facilidade de uso, desempenho, flexibilidade e suporte da comunidade para atender às necessidades da sua equipe.

Otimização ao longo do tempo

GAMEOPS06: Como você otimiza seu jogo ao longo do tempo?

Otimize seu jogo ao longo do tempo monitorando as principais métricas e dados de telemetria para identificar tendências dos jogadores, desempenho do sistema e áreas de melhoria. Atualize continuamente o design, a infraestrutura e a abordagem de teste de carga do jogo com base nesses

insights, enquanto se adapta às novas tecnologias e estruturas para fornecer desempenho e experiência de jogador ideais à medida que o jogo evolui.

Práticas recomendadas

- GAMEOPS06-BP01 Monitore as principais métricas do jogo para identificar tendências e padrões dos jogadores e use as informações para melhorar o jogo
- GAMEOPS06-BP02 Atualize e adapte a abordagem de teste de carga conforme o jogo muda

GAMEOPS06-BP01 Monitore as principais métricas do jogo para identificar tendências e padrões dos jogadores e use as informações para melhorar o jogo

Além do uso do sistema cliente do jogo, do uso do aplicativo, das exceções e dos dados de falhas, capture os dados de telemetria do jogo que são enviados para o sistema de back-end do jogo. Esses dados devem representar a atividade do jogador para que você possa entender como os jogadores interagem com os vários recursos do jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Dependendo de sua implementação, os clientes do jogo podem coletar dados de telemetria em recursos ou locais predefinidos do jogo em um mundo de jogo. Os dados são enviados ao serviço de ingestão de back-end para processamento. Se o serviço de back-end estiver inacessível, os clientes poderão armazenar os dados localmente no dispositivo local até que o serviço de back-end esteja disponível novamente. Os designers de jogos usam esses dados de telemetria para analisar como os jogadores estão jogando e se há anomalias no jogo.

Por exemplo, os movimentos e interações dos jogadores com itens em um mapa podem ser extraídos dos dados de telemetria e plotados como um mapa térmico das atividades dos jogadores no jogo em uma janela de tempo definida. Esses dados ajudam os designers do jogo a identificar a necessidade de equilibrar vários elementos do jogo, como o poder de uma arma, o poder de um personagem no jogo ou a complexidade de um mapa. Os dados brutos de telemetria geralmente são armazenados e processados para extrair análises que podem ser visualizadas pelos analistas.

A implementação da AnalyticsPipeline solução Game ajuda os desenvolvedores de jogos a lançar um pipeline de dados escalável sem servidor para ingerir, armazenar e analisar dados de

telemetria gerados a partir de jogos e serviços. A solução suporta a ingestão de dados por streaming, permitindo que os usuários obtenham informações sobre seus jogos e outros aplicativos em minutos.

Para ingestão, armazenamento, processamento e análise de dados de telemetria de jogos personalizados, AWS também oferece vários serviços especializados para processamento e análise de big data.

Etapas de implementação

- Capture dados de telemetria do jogo: colete dados sobre a atividade do jogador, o uso do sistema, exceções e falhas para entender as interações dos jogadores e identificar problemas.
- Implemente a coleta de telemetria: use recursos ou locais predefinidos do jogo para coletar dados de telemetria e enviá-los aos serviços de back-end, armazenando-os localmente se o back-end estiver inacessível.
- Use soluções de AWS análise: use AWS serviços como o Game Analytics Pipeline para ingestão, armazenamento e análise de dados escaláveis, bem como serviços especializados de processamento e análise de big data.

GAMEOPS06-BP02 Atualize e adapte a abordagem de teste de carga conforme o jogo muda

A otimização da abordagem de teste de carga é um processo contínuo que deve evoluir junto com o ciclo de desenvolvimento do jogo. À medida que o jogo cresce em complexidade, base de usuários e conjunto de recursos, a estratégia de teste de carga deve se adaptar para verificar se simula com precisão as condições do mundo real e fornece informações acionáveis.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Considere o seguinte:

Cenários de teste ausentes ou desatualizados

À medida que novas funcionalidades são adicionadas a um jogo durante o processo de desenvolvimento, crie e execute novos cenários de teste de carga para validar o desempenho e a escalabilidade dos novos recursos. Da mesma forma, recursos e funcionalidades geralmente são refatorados para melhorar o desempenho, atender ao feedback dos jogadores ou se alinhar às

novas metas de design, exigindo que os cenários de teste sejam atualizados continuamente para acompanhar as mudanças e realmente testar e refletir o estado do sistema.

Novas estruturas de teste de carga

Talvez os desenvolvedores precisem alterar as estruturas de teste de carga por vários motivos:

- A estrutura inicial pode não ser mais capaz de simular adequadamente a carga do usuário ou fornecer o nível necessário de visão sobre o desempenho do sistema
- Novos recursos do jogo podem exigir suporte a testes de carga para novos APIs protocolos ou pontos de integração
- Os desenvolvedores podem desejar recursos mais avançados à medida que se sentem mais confortáveis com o processo de teste de carga
- Preferência por estruturas que se alinhem melhor à experiência técnica, às linguagens de programação ou às cadeias de ferramentas existentes da equipe

Ao avaliar e adaptar cuidadosamente ao longo do tempo, os desenvolvedores podem alinhar o processo de teste de carga às mudanças nos requisitos do jogo e continuar fornecendo as informações necessárias para otimizar e melhorar a experiência geral do usuário.

Otimizando o custo

A facilidade e a conveniência de usar AWS serviços gerenciados podem ser altamente benéficas, especialmente nos estágios iniciais de desenvolvimento. Esses serviços abstraem o gerenciamento da infraestrutura subjacente, permitindo que as equipes configurem rapidamente sua solução e se concentrem exclusivamente na criação de cenários de teste de carga e na análise dos resultados. No entanto, o uso de serviços gerenciados geralmente pode ter um custo maior devido ao valor adicional e à conveniência que eles oferecem, como provisionamento, configuração e manutenção da infraestrutura, além de fornecer recursos de alta disponibilidade, escalabilidade e monitoramento.

À medida que as equipes amadurecem e ficam mais confortáveis e confiantes com o processo de teste de carga, pode chegar um momento em que o autogerenciamento da infraestrutura possa oferecer otimização adicional e economia de custos. Embora essa abordagem prática aumente a sobrecarga operacional, ter controle direto sobre os recursos computacionais, as configurações, os comportamentos de escalabilidade e a utilização dos recursos pode abrir novas oportunidades de ajuste fino e redução de custos. Por exemplo, pode fazer sentido que as equipes iniciem sua jornada de teste de carga com uma arquitetura AWS Fargate sem servidor e, posteriormente, passem para o autogerenciamento dos nós subjacentes em um cluster do Amazon EKS.

Etapas de implementação

- Atualize cenários de teste: crie e atualize continuamente cenários de teste de carga para validar novos recursos e funcionalidades refatoradas e verificar se eles refletem o estado atual do jogo.
- Avalie as estruturas de teste de carga: adapte-se às novas estruturas conforme necessário para simular a carga do usuário, oferecer suporte a novos protocolos e alinhar-se à experiência e às cadeias de ferramentas da equipe.
- Otimize os custos: comece com AWS serviços gerenciados para facilidade e conveniência e, em seguida, considere a infraestrutura autogerenciada para reduzir os custos à medida que a equipe se sentir mais confortável com o processo de teste de carga.

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas relacionadas à excelência operacional.

Documentação e blogs

- Melhores práticas de arquitetura para tecnologia de jogos
- Gerenciando seu estúdio de jogos no AWS pt.1
- Gerenciando seu estúdio de jogos no AWS pt. 2
- Gerenciando seu estúdio de jogos no AWS pt. 3
- Estabelecendo seu AWS ambiente de melhores práticas
- estratégia de várias contas para sua zona de pouso da Control Tower
- Pipeline de análise de jogos
- Maximize seus insights de dados de jogos com o pipeline de análise de jogos
- Aproveitando o AWS Glue Amazon Redshift Spectrum para insights de jogadores
- Como configurar um CI/CD pipeline no
- Como a Good Job Games acelera 43% com AWS o Build Pipeline
- Implementando um pipeline de construção para aplicativos móveis Unity
- Outros CI/CD blogs relevantes
- Jogo DevOps facilitado com o blog Game-Server CD Pipeline
- A Harmony Games implanta um back-end de jogo totalmente personalizado utilizando AWS Cloud Development Kit (AWS CDK) ()AWS CDK

- GameLiftPrepare-se para o lançamento
- Hospedagem híbrida de servidores de jogos com o Amazon Gamelift Anywhere
- Acelere o desenvolvimento de servidores de jogos com o Amazon Gamelift Anywhere e o Amazon Gamelift Agent
- Como hospedar seu jogo Unreal Engine por menos de $1 por jogador com o Amazon Gamelift
- Nova orientação de solução para criar back-ends de jogos multiplataforma escaláveis em AWS
- Teste de carga do mecanismo de jogo de back-end Pragma para 1 milhão de usuários simultâneos em AWS
- Como a Code Wizards testou Nakama do Heroic Lab para dois milhões de jogadores simultâneos com AWS
- Melhores práticas com unidades organizacionais
- AWS X-Ray
- AWS Contagem regressiva
- AWS para Games Solutions Hub

Soluções para parceiros

- New Relic
- APM do Splunk
- Backtrace.io
- Sentinela
- APM do Datadog
- Honeycomb.io

Whitepapers

- Organizando seu ambiente usando várias contas
- Introdução aos padrões escaláveis de desenvolvimento de jogos em AWS

Vídeos

- YouTubesérie: Building Games on AWS

- AWS para jogos: Boss LEVEL Podcast
- Re: Invent 2023: ampliando a escala AWS para os primeiros 10 milhões de usuários
- Re: Invent 2022: Como a Riot Games processa 20 TB de análises diariamente em AWS
- Re: Invent 2022: Como a Riot Games criou um AWS mecanismo de geração de relatórios de governança
- Re: Invent 2023: Implementando padrões de design distribuídos em AWS
- Re:Invent 2023: escalando um jogo multijogador para milhões com Mortal Kombat 1
- Re: Invent 2022: A evolução da engenharia do caos na Netflix
- Re: Invent 2023: melhores práticas para governança de nuvem
- Re: Invent 2023: melhores práticas para criar arquiteturas multirregionais em AWS

Materiais de treinamento

- Currículo — Começando com AWS for Games, Parte 1

Segurança

O pilar de segurança inclui a capacidade de proteger informações, sistemas e ativos e, ao mesmo tempo, agregar valor comercial por meio de avaliações e mitigação de riscos. Devido à visibilidade global e ao grande número de jogadores, os jogos são um alvo desejável para exploradores, hackers e outros que procuram maneiras de explorar e abusar dos sistemas. Isso geralmente pode resultar em uma experiência decepcionante para o jogador e em custos aumentados para o desenvolvedor do jogo, se não houver uma base de segurança sólida estabelecida.

Conforme descrito no Modelo de Responsabilidade Compartilhada, é importante entender quais aspectos da segurança são de responsabilidade do cliente AWS e quais aspectos são de responsabilidade do cliente, para que você esteja preparado para manter uma postura de segurança sólida. Esse pilar fornece orientações sobre as melhores práticas de segurança na nuvem para você considerar ao desenvolver e operar jogos na nuvem.

Antes de arquitetar um sistema, você deve estabelecer um conjunto de melhores práticas de segurança que inclua controles de acesso. Além disso, você deve ser capaz de identificar incidentes de segurança e proteger seus sistemas e serviços, mantendo a confidencialidade e a integridade dos dados por meio da proteção de dados. Você deve ter um processo bem definido e treinado para responder a incidentes de segurança. Essas ferramentas e técnicas são importantes porque apoiam os objetivos de negócios, como evitar perdas financeiras ou cumprir obrigações regulatórias.

Exemplo de cliente

AnyCompany Games é um estúdio de jogos fictício que está melhorando sua postura de segurança. A segurança pode ser fácil de entender quando há uma explicação de sua aplicação direta. AnyCompany Os jogos são usados nesta seção para contextualizar as melhores práticas de segurança descritas no pilar

Áreas de foco

- Princípios de design
- Fundamentos de segurança
- Segurança contínua
- Gerenciamento de identidade e acesso
- Controle de acesso
- Detecção

- Proteção da infraestrutura
- Resposta a incidentes
- Segurança de aplicações
- Automatize a segurança
- Modelagem de ameaças
- Recursos

Princípios de design

Além dos princípios de design do pilar de segurança do whitepaper Well-Architected Framework, o princípio de design a seguir pode fortalecer a segurança da carga de trabalho do seu jogo na nuvem:

- Monitore e modere o comportamento de uso do jogador: capture e analise dados de uso para entender como os jogadores interagem com seu jogo e seus recursos sociais. Ao analisar esses dados, você pode detectar e responder a comportamentos abusivos e inadequados que podem degradar a experiência do jogador.

Fundamentos de segurança

GAMESEC01: Como você implementa os fundamentos de segurança para o desenvolvimento de jogos?

Os estúdios de jogos exigem uma abordagem de segurança exclusiva que proteja tanto os ambientes de desenvolvimento quanto os serviços de jogadores ao vivo. Uma estratégia AWS de segurança robusta para estúdios de jogos requer três componentes interconectados: uma estrutura de várias contas, autenticação forte e uma estratégia de autorização clara usando políticas do IAM. Uma AWS estrutura de várias contas permite que os estúdios separem diferentes projetos de jogos, estágios de desenvolvimento e ambientes de ferramentas. Isso dá aos estúdios um controle mais granular para coisas como acesso a ambientes ou serviços específicos. A ativação da autenticação forte garante que os membros da equipe possam acessar com segurança os recursos de desenvolvimento, trabalhando no estúdio ou remotamente, mantendo controles rígidos sobre o código-fonte, as compilações de jogos e as ferramentas proprietárias. Os estúdios também devem ter uma estratégia de autorização clara para conceder permissões usando o

princípio do menor privilégio com permissões e funções do IAM. Use as funções do IAM para atribuir permissões entre diferentes funções da equipe de desenvolvimento, como dar às equipes de desenvolvimento acesso a AWS serviços de baixo nível e restringir artistas e designers a sistemas específicos de gerenciamento de ativos e criação. Essa abordagem especializada garante que os estúdios de jogos possam proteger sua propriedade intelectual, manter fluxos de trabalho de desenvolvimento eficientes e escalar suas equipes com segurança, ao mesmo tempo em que oferece aos desenvolvedores o acesso adequado para iterar rapidamente em seus projetos.

Práticas recomendadas

- GAMESEC01-BP01 Use funções e acesso federado, em vez do usuário raiz da conta, para realizar ações em seu ambiente AWS
- GAMESEC01-BP02 Use AWS Control Tower para configurar rapidamente um ambiente de várias contas em AWS
- GAMESEC01-BP03 Use políticas de função de privilégio mínimo adaptadas a funções de trabalho específicas
- GAMESEC01-BP04 Use funções e políticas de acesso federado junto com políticas de acesso em nível de conta para conceder acesso aos seus recursos AWS
- GAMESEC01-BP05 Use um provedor de identidade central

GAMESEC01-BP01 Use funções e acesso federado, em vez do usuário raiz da conta, para realizar ações em seu ambiente AWS

Ao criar um Conta da AWS, você começa com uma identidade conhecida como usuário root, que é acessada usando o endereço de e-mail e a senha associados à conta. O usuário root tem acesso completo aos AWS serviços e recursos dessa conta. Na maioria dos casos, você deve evitar usar o usuário root para day-to-day tarefas. Quando o acesso no nível raiz for necessário, confirme se é absolutamente necessário e verifique se há registros e grades de proteção adicionais para rastrear seu uso.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Em uma AWS Organizations configuração, cada conta ainda tem seu próprio usuário raiz, mas o day-to-day acesso deve ser gerenciado por meio de funções do IAM e usuários do IAM Identity Center. Crie acesso baseado em funções adaptado às etapas e equipes do ciclo de vida do seu jogo.

Por exemplo, a equipe de operações ao vivo pode precisar de permissões para gerenciar eventos no jogo, enquanto os desenvolvedores precisam acessar atualizações push. Ao trabalhar com serviços ou parceiros de terceiros, use o acesso federado para permitir uma colaboração segura sem expor a infraestrutura confidencial. Essa abordagem verifica se cada usuário ou parceiro tem apenas o acesso de que precisa, mantendo a segurança da infraestrutura e dos dados do jogador do seu jogo.

Exemplo de cliente

AnyCompany Os jogos implementaram o controle de acesso baseado em funções ao desenvolver seu novo jogo. Ao usar funções específicas do IAM para suas diversas equipes de desenvolvimento, elas evitam usar credenciais compartilhadas. Essa configuração permite que uma equipe de desenvolvimento assuma uma função nos sistemas principais do jogo, enquanto a função da equipe de conteúdo só pode acessar os serviços de gerenciamento de ativos.

Etapas de implementação

- Não use o usuário root depois de configurar uma conta, a menos que seja absolutamente necessário. Crie a conta, proteja o usuário raiz e crie imediatamente as funções administrativas do IAM necessárias e atribua essa função ao usuário federado.
- Use o usuário root somente quando precisar realizar um número limitado de tarefas que só estão disponíveis para o usuário root. Exemplos dessas tarefas incluem alterar seu endereço de e-mail de usuário raiz e alterar seu plano de AWS suporte.

GAMESEC01-BP02 Use AWS Control Tower para configurar rapidamente um ambiente de várias contas em AWS

Se você começar a usar AWS com apenas uma única conta, poderá descobrir que seu estúdio de jogos cresce à medida que o processo de desenvolvimento de jogos avança. Por exemplo, com um único Conta da AWS, você pode começar a atingir os limites de serviço ou seus custos para diferentes projetos e cargas de trabalho podem se tornar mais complexos. A criação de contas diferentes para diferentes títulos e ambientes de jogos permite que as equipes experimentem novos recursos, contornem os limites do serviço e mantenham a postura de segurança e a conformidade. Ao implementar uma estratégia de várias contas em AWS, você pode se beneficiar da distribuição dos limites de serviço em várias contas e obter informações sobre seus AWS custos.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

É um equívoco comum pensar que usar vários Contas da AWS será automaticamente mais confuso e demorado. Em vez disso, usar AWS serviços projetados para facilitar a governança de várias contas pode ajudar seu estúdio de jogos a gastar menos tempo gerenciando suas contas.

Você pode usar AWS Control Tower é um serviço para provisionar com segurança um ambiente com várias AWS contas. O Control Tower é recomendado se você estiver construindo um novo AWS ambiente, iniciando sua jornada ou se for completamente novo AWS. AWS Durante o breve processo de configuração, você pode se integrar a outros AWS serviços envolvidos no gerenciamento de contas e acesso de usuários AWS Organizations, como Service Catalog e AWS IAM Identity Center.

Exemplo de cliente

AnyCompany Inicialmente Conta da AWS, os jogos operavam a partir de um único e enfrentaram vários obstáculos quando uma das equipes de desenvolvimento de seus jogos atingiu os limites EC2 de serviço durante um teste beta crucial. Ao mesmo tempo, sua equipe de desenvolvimento de um jogo diferente teve dificuldades com a alocação de recursos para seu pipeline de testes automatizados. A situação chegou a um ponto de ruptura quando a AnyCompany Games não conseguiu separar com precisão os custos entre os projetos, dificultando o orçamento para o desenvolvimento de cada jogo.

AnyCompany Em seguida, os jogos implementaram uma estratégia de várias contas usando AWS Control Tower. Eles criaram contas separadas para cada projeto de jogo, com ambientes distintos de desenvolvimento, controle de qualidade e produção. Essa separação em nível de conta isola os dados e ativos de cada projeto, para que as equipes que trabalham em um jogo não possam acessar ou modificar recursos de outro. Por meio AWS Organizations disso, eles estabeleceram uma estrutura de cobrança centralizada que mostrava claramente os custos de infraestrutura de cada jogo e também criaram políticas de acesso em toda a organização.

Etapas de implementação

- Use a torre AWS de controle para configurar um ambiente automatizado de várias contas.
- Organize contas com base em ambientes (como desenvolvimento, controle de qualidade e produção).
- Use o AWS IAM Identity Center e o Service Catalog para centralizar as permissões do usuário e simplificar o provisionamento de recursos em todas as contas.

GAMESEC01-BP03 Use políticas de função de privilégio mínimo adaptadas a funções de trabalho específicas

Configurar políticas do IAM é uma parte essencial do estabelecimento de uma base de segurança sólida. 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. Por exemplo, as equipes de controle de qualidade precisam ter acesso para mudar as coisas nos ambientes de teste, mas não devem ter a capacidade de modificar o ambiente de produção.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Você pode começar com permissões amplas, como políticas gerenciadas, enquanto explora as permissões necessárias para sua carga de trabalho ou caso de uso. À medida que seu caso de uso se desenvolve, você pode trabalhar para reduzir as permissões que concede para caminhar em direção ao privilégio mínimo.

Etapas de implementação

- Siga a prática de permissões com privilégios mínimos para criar funções do IAM para usuários e aplicativos.
- Use políticas AWS gerenciadas para fornecer acesso amplo rapidamente e, ao mesmo tempo, identificar as permissões específicas que as equipes ou os aplicativos precisam para realizar suas tarefas.
- Os estúdios também podem usar a geração de políticas do IAM access analyzer para gerar políticas personalizadas do IAM com base em CloudTrail eventos que identificam ações e serviços usados por uma entrada do IAM.
- Analise regularmente as políticas do IAM e edite políticas excessivamente permissivas.

GAMESEC01-BP04 Use funções e políticas de acesso federado junto com políticas de acesso em nível de conta para conceder acesso aos seus recursos AWS

AWS Os novos usuários geralmente usam políticas do IAM somente ao conceder acesso a outras pessoas. No entanto, se você estiver usando AWS Organizations, considere como usar as políticas

de controle de serviço junto com as políticas do IAM para conceder aos membros da equipe e contratados do estúdio os níveis de acesso necessários.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Você pode criar políticas do IAM para permitir ou negar acesso a AWS serviços ou ações de API que funcionam com AWS Identity and Access Management. Eles só podem ser aplicados às identidades do IAM, como usuários, grupos ou funções. Por exemplo, uma política do IAM pode ser usada para fornecer ao usuário acesso somente de leitura ao Amazon S3.

As políticas de controle de serviço (SCPs) são proteções para você. Contas da AWS Um SCP não concede permissões, elas são usadas para restringir ações em AWS serviços para contas de membros individuais. Por exemplo, um SCP pode impedir o acesso Conta da AWS a uma região específica.

Quando uma ação é tomada, a política relevante do IAM é avaliada em combinação com SCPs o. Seguindo o exemplo anterior, se uma função tentar executar uma EC2 instância, o IAM indica se ela é permitida (“Permitir” para ec2:RunInstances) e SCPs determinará se sua escolha de região é válida (“us-east-1" é permitido, mas “us-west-1" é negado pelo SCP).

Coloque as políticas do IAM em camadas e SCPs possa verificar se qualquer pessoa que acesse seus AWS recursos receberá apenas as permissões apropriadas de que precisa. É especialmente importante considerar isso se seus recursos Contas da AWS e recursos abrangem várias regiões, mas nem todos em seu estúdio de jogos precisam acessar todas elas.

Você pode personalizar as políticas do IAM para conceder permissões específicas a equipes específicas para atualizar coisas como configurações de jogos, gerenciar dados de jogadores, configurar eventos promocionais e moderar conteúdo gerado pelo usuário. Enquanto isso, você pode usar SCPs para aplicar controles em toda a organização, cruciais para as operações do jogo. Isso pode incluir restringir a implantação somente às regiões aprovadas em que o jogo opera, ajudar a impedir o acesso não autorizado a armazenamentos de dados confidenciais de jogadores, impor requisitos de conformidade e controlar custos limitando o uso do serviço nas contas de desenvolvimento.

Etapas de implementação

- Use as políticas do IAM para gerenciar permissões para usuários, grupos ou funções individuais.

- Use políticas de controle de serviço (SCPs) AWS Organizations para impor permissões em nível de conta.
- Combine as políticas do IAM e SCPs conceda somente o acesso necessário para usuários e contas específicos.

Recursos

- Políticas e permissões no AWS IAM
- Políticas de controle de serviço
- Políticas gerenciadas pela AWS para funções de trabalho

GAMESEC01-BP05 Use um provedor de identidade central

Um provedor de identidade central atua como uma fonte única para armazenar e gerenciar credenciais, identidades, permissões e autenticação do usuário.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Use um provedor de identidade central para agilizar seu processo de autenticação de usuários, aplicar políticas de segurança consistentes e simplificar o gerenciamento de usuários em seus Contas da AWS aplicativos. Ter uma abordagem centralizada elimina a necessidade de gerenciar as identidades e credenciais dos usuários separadamente, o que reduz o risco de inconsistências, redundâncias e outras vulnerabilidades de segurança. A consolidação das identidades e da autenticação dos usuários em um só lugar também permite melhor visibilidade, controle e auditabilidade em todo o seu ambiente. AWS

Exemplo de cliente

AnyCompany Os jogos enfrentaram desafios significativos ao gerenciar o acesso dos desenvolvedores em sua AWS infraestrutura em rápida expansão. Sua equipe de desenvolvimento cresceu de 50 para 200 pessoas em três títulos principais. Inicialmente, cada equipe do projeto gerenciava suas próprias credenciais de AWS acesso, resultando em práticas de segurança inconsistentes, atrasos na integração de novos desenvolvedores e incidentes de segurança ocasionais.

O estúdio implementou o AWS IAM Identity Center como seu provedor central de identidade, consolidando o gerenciamento de usuários em um único sistema. Eles o conectaram ao diretório corporativo existente, permitindo que os desenvolvedores usassem as mesmas credenciais da empresa para AWS acesso. Agora, os desenvolvedores usam seu login único e existente da empresa para obter o AWS acesso necessário para concluir seu trabalho

Etapas de implementação

- Considere usar o AWS IAM Identity Center como seu provedor de identidade central. Isso fornece gerenciamento de acesso consistente em toda a sua empresa Contas da AWS, fornece aos seus funcionários a autenticação de login único e simplifica a auditoria de acesso do usuário aos seus aplicativos. AWS O IAM Identity Center também se conecta às identidades corporativas existentes dos provedores de identidade compatíveis.

Segurança contínua

GAMESEC02: Como você alcança, mantém e monitora as melhores práticas de segurança contínuas?

Aderir às melhores práticas de segurança é fundamental para empresas de todos os setores, mas especialmente na indústria de jogos. A indústria de jogos depende de cultivar e manter a confiança dos jogadores e criar uma reputação forte, e até mesmo pequenos problemas de segurança podem rapidamente minar essa confiança.

Além disso, a natureza global da indústria de jogos exige conformidade com vários regulamentos e padrões do setor que regem a proteção de dados, a privacidade do consumidor e a segurança nas regiões em que os jogos são oferecidos. A jogabilidade justa e segura é outro aspecto crítico que ressalta a importância de medidas de segurança robustas. Trapacear, hackear e outras formas de exploração do jogo podem atrapalhar a experiência de jogo de jogadores legítimos, o que torna essenciais controles de segurança fortes para manter a integridade da jogabilidade e promover condições equitativas para os participantes.

Práticas recomendadas

- GAMESEC02-BP01 Use modelos prontos para implantar para práticas de segurança padrão

- GAMESEC02-BP02 Use técnicas automatizadas de remediação quando surgir um evento de segurança

GAMESEC02-BP01 Use modelos prontos para implantar para práticas de segurança padrão

Ready-to-deployos modelos fornecem uma forma proativa e ágil de avaliar sua postura de segurança na nuvem. Modelos pré-configurados avaliam sua segurança na nuvem e implementam as mudanças necessárias imediatamente. Os modelos abrangem uma ampla variedade de melhores práticas em várias tecnologias e estruturas de segurança amplamente aceitas. O uso de modelos pode ajudar os estúdios de jogos a manter configurações de infraestrutura consistentes, especialmente porque eles podem escalar e adicionar mais Contas da AWS para suportar novas cargas de trabalho.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Ao usar AWS serviços e implementar ready-to-deploy modelos, os desenvolvedores de jogos podem avaliar e fortalecer proativamente sua postura de segurança na nuvem, protegendo sua propriedade intelectual, protegendo os dados dos jogadores e promovendo um cenário de jogos seguro por meio de avaliações regulares de segurança e monitoramento contínuo para identificar e resolver prontamente possíveis vulnerabilidades.

Exemplo de cliente

AnyCompany Os jogos enfrentaram um desafio significativo ao se preparar para lançar seu próximo título na indústria europeia. Eles perceberam que suas práticas existentes de tratamento de dados não atendiam aos requisitos do GDPR. Eles recorreram à AWS Security Hub CSPM AWS Config e seus ready-to-deploy modelos em busca de uma solução. A equipe implementou o pacote de conformidade específico do GDPR AWS Config, que avaliou automaticamente sua infraestrutura existente em relação aos padrões do GDPR. Essa verificação inicial revelou várias lacunas críticas, como políticas inadequadas de retenção de dados e controles de acesso inadequados sobre onde os dados do jogador foram armazenados. Usando as regras predefinidas do modelo, a AnyCompany Games implementou rapidamente as mudanças necessárias. Além disso, as verificações de conformidade automatizadas contínuas fornecidas pelo modelo permitiram que a pequena equipe mantivesse a conformidade com o GDPR sem esforço, mesmo enquanto continuava atualizando e expandindo o jogo.

Etapas de implementação

- Use modelos para práticas de segurança padrão, como regras gerenciadas e pacotes de conformidade AWS Config e padrões em. AWS Security Hub CSPM
- Analise os detalhes dos padrões CSPM do Security Hub para determinar quais estão mais alinhados às necessidades de segurança do seu estúdio de jogos.

GAMESEC02-BP02 Use técnicas automatizadas de remediação quando surgir um evento de segurança

Usando técnicas automatizadas de remediação, os desenvolvedores de jogos podem proteger e manter proativamente sua infraestrutura de jogos e minimizar o impacto potencial que um incidente de segurança possa ter. Se um problema de segurança for detectado, use um runbook para orientar sua resposta à situação. Automatize essas respostas sempre que possível para corrigir problemas mais rapidamente e reduzir seu impacto. Isso melhora a experiência do jogador ao reduzir a chance de tempo de inatividade e interrupções no jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

A preparação para responder aos problemas de segurança não apenas protege a experiência dos jogadores, mas também atende aos vários padrões regulatórios e de conformidade. Além disso, o uso de respostas de segurança automatizadas expande suas operações de segurança à medida que suas cargas de trabalho se expandem. AWS fornece serviços para ajudar a identificar e automatizar uma resposta a esses incidentes.

Exemplo de cliente

AnyCompany Os jogos enfrentaram um incidente crítico de segurança quando um bucket S3 contendo modelos e texturas de personagens inéditos para seu próximo jogo foi acidentalmente divulgado durante uma atualização rotineira do pipeline de ativos. O sistema de segurança automatizado detectou a alteração na permissão do bucket poucos minutos após a modificação. O sistema executou imediatamente seu runbook de remediação: revertendo o bucket para o status privado, registrando tentativas de acesso durante a janela de exposição, notificando a equipe de segurança e criando um CloudTrail registro detalhado das alterações de permissão.

Etapas de implementação

- Use a AWS solução Automated Security Response on para implementar runbooks de automação que definem as ações que serão tomadas automaticamente em resposta aos eventos de segurança em AWS Security Hub CSPM.

Recursos

- AWS for Games Blog — Gerenciando seu estúdio de jogos em AWS: Parte 1
- AWS for Games Blog — Gerenciando seu estúdio de jogos na AWS parte 2
- Registre uma unidade organizacional existente com AWS Control Tower
- Conta da AWS usuário root
- Tarefas que exigem credenciais de usuário root
- Resposta de segurança automatizada em AWS

Gerenciamento de identidade e acesso

GAMESEC03: Como você gerencia a identidade do jogador e o gerenciamento de acesso?

Ao desenvolver um jogo, você deve determinar como fornecerá aos jogadores acesso ao seu jogo e aos serviços relacionados. A seção a seguir explora as considerações de design, incluindo autenticação, autorização e autenticação multifatorial do jogador.

Práticas recomendadas

- GAMESEC03-BP01 Determine sua abordagem para identificar e controlar o acesso do jogador ao ambiente e aos recursos do seu jogo
- GAMESEC03-BP02 Autenticar solicitações enviadas ao serviço de back-end do seu jogo
- GAMESEC03-BP03 Use seu serviço de back-end de jogos para validar as solicitações dos jogadores para participar de um jogo multijogador
- GAMESEC03-BP04 Aplique uma política de segurança rígida para contas de usuário de jogadores, exigindo uma senha forte
- GAMESEC03-BP05 Ofereça uma opção para os jogadores configurarem a autenticação multifator (MFA) em suas contas

GAMESEC03-BP01 Determine sua abordagem para identificar e controlar o acesso do jogador ao ambiente e aos recursos do seu jogo

Essa decisão é influenciada pela estratégia de aquisição e monetização de jogadores, pela experiência do jogador e por outros fatores, como os recursos existentes que podem ser fornecidos por seus parceiros de publicação de jogos. Por exemplo, um jogo pode exigir compras e exigir que o jogador crie um perfil de usuário para associar formas de pagamento em dinheiro real à sua conta.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Como alternativa, um jogo pode querer reduzir a barreira de entrada para experiências de jogadores iniciantes, eliminando a necessidade de criar uma conta de usuário antes de jogar, aumentando assim a chance de um jogador experimentar o jogo pela primeira vez. Normalmente, os jogos implementam uma ou mais combinações de abordagens de gerenciamento de acesso e identidade do jogador para o jogo.

Acesso não autenticado ou anônimo

Esse nível de acesso é útil em situações em que um jogo não exige que o jogador crie uma nova conta de usuário ou vincule sua identidade em redes sociais e sistemas de jogos. Essa é a maneira mais simples e rápida de um jogador começar a jogar e é particularmente útil em jogos para celular, nos quais um desenvolvedor de jogos pode querer reduzir a barreira de entrada para a experiência inicial.

Nesse cenário de acesso, se quiser identificar o uso da instalação do jogo, você pode programar o cliente do jogo para gerar e armazenar um identificador exclusivo no dispositivo do jogador. Esse identificador exclusivo é usado para identificar o jogador em todas as sessões de jogo em seu dispositivo e permitir relatórios analíticos sobre o uso ao longo do tempo. Posteriormente, se um jogador optar por criar uma conta, você poderá associar a nova conta de usuário ao identificador exclusivo gerado anteriormente. Isso vinculará sua nova identidade de jogador ao seu uso histórico, que pode incluir estatísticas e conquistas do jogo.

Se um jogador eventualmente não criar e vincular uma conta, o dispositivo que o jogador usa para interagir com o jogo pode ser identificado de forma exclusiva, mas as informações recuperáveis sobre o jogador não são coletadas e armazenadas. Portanto, se o player quebrar ou perder o dispositivo, os dados armazenados anteriormente associados ao dispositivo também serão perdidos e poderão não ser recuperáveis.

Autenticação com nome de usuário e senha

Um jogo pode permitir que os jogadores criem suas próprias contas de usuário com um nome de usuário e senha armazenados no backend do jogo. Isso pode ocorrer quando um desenvolvedor de jogos está colaborando com um editor de jogos que já tem um sistema de contas de jogador existente ao qual o desenvolvedor pode se integrar. Como alternativa, um desenvolvedor que publica seus próprios jogos pode querer simplificar a experiência do jogador, permitindo que os jogadores criem uma única conta de usuário para acessar todos os jogos que publicam.

Autenticação e vinculação de contas com redes sociais e sistemas de jogos de terceiros

É comum que jogos online e jogos com recursos sociais forneçam federação de provedores de identidade terceirizados para simplificar a experiência do jogador. Em vez de pedir aos jogadores que criem uma combinação de nome de usuário e senha para se autenticar, você pode usar a federação de identidades para permitir que os jogadores se autentiquem usando suas contas de terceiros em redes sociais e sistemas de jogos. Esse processo de login simplifica a experiência de login e registro dos jogadores. Ele também fornece uma alternativa conveniente à criação obrigatória de contas e um método simples para os jogadores acessarem os jogos.

Para desenvolvedores de jogos, um processo de login federado pode oferecer um fluxo de trabalho simplificado de verificação de jogadores. Também pode fornecer uma maneira mais confiável de gerenciar os dados do jogador que são usados para personalização. Isso ocorre porque você não precisa pedir aos jogadores que forneçam determinados dados que eles provavelmente já forneceram ao provedor de identidade terceirizado. Além disso, esses sistemas oferecem integração com recursos sociais adicionais, como a capacidade de vincular jogadores a seus amigos.

Etapas de implementação

- Use o acesso não autenticado ou anônimo para reduzir as barreiras para jogadores iniciantes, gerando um identificador de dispositivo exclusivo para rastrear o uso e habilitar a vinculação de contas posteriormente.
- Implemente a autenticação de nome de usuário e senha para contas de usuário dedicadas, usando sistemas de contas de jogadores existentes ou criando uma experiência unificada em todos os jogos.
- Integre provedores de identidade terceirizados para autenticação federada, simplificando os processos de login e permitindo o acesso a recursos sociais e dados de personalização.

GAMESEC03-BP02 Autenticar solicitações enviadas ao serviço de back-end do seu jogo

Solicitações de autenticação enviadas ao serviço de back-end do jogo podem impedir que solicitações indesejadas sejam bem-sucedidas.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Você deve fornecer um serviço de autenticação para que os jogadores façam login, que deve devolver tokens seguros de curta duração, como um JSON Web Token (JWT), ao cliente do jogo quando um jogador se autentica com sucesso.

Esses tokens podem incluir declarações de reivindicações que contêm atributos do jogador e outros metadados relevantes. Esses metadados relevantes podem ser usados em solicitações subsequentes enviadas do cliente do jogo para o back-end do jogo para autenticar solicitações e autorizá-las no contexto do jogador autenticado.

Você tem a opção de projetar e criar seu próprio sistema de autenticação de jogadores, o que exigiria melhoria e manutenção contínuas, ou você pode usar os recursos escaláveis e seguros de cadastro, login e controle de acesso de usuários fornecidos pelo Amazon Cognito.

Os grupos de usuários do Amazon Cognito incluem um diretório de usuários para autenticação e autorização. Um grupo de usuários permite APIs que você se integre ao seu jogo para fluxos de trabalho de inscrição, login e redefinição de senha, que podem ser integrados a provedores de identidade terceirizados. Os Application Load Balancers e o Amazon API Gateway fornecem integrações com o Cognito para integrar a autenticação do usuário para solicitações enviadas aos seus back-ends de jogos personalizados hospedados com esses serviços.

Se seu jogo suporta acesso anônimo e você não consegue autenticar um jogador, você pode usar uma abordagem de autenticação de cliente para fornecer uma experiência mais segura ao se integrar ao back-end do jogo. Se seu cliente de jogo usa AWS serviços diretamente, as solicitações para esses serviços devem ser assinadas usando credenciais. Para fornecer credenciais ao seu cliente de jogo para usuários não autenticados, você pode usar o AWS SDK para recuperar credenciais de curta duração dos grupos de identidade do Amazon Cognito que podem ser usadas para assinar suas solicitações aos serviços. AWS Essas credenciais podem ser atualizadas no seu cliente de jogo.

Além de se integrar diretamente ao AWS SDK a partir do cliente do jogo, você também pode criar seu próprio back-end de jogos usando um serviço como o Amazon API Gateway, que oferece suporte à autorização personalizada. Ao criar seu próprio serviço de back-end de jogos, você pode obter controle autoritário sobre as solicitações com uma lógica personalizada do lado do servidor.

Para obter mais informações sobre como criar um serviço de back-end para jogos hospedados usando a Amazon GameLift, consulte Projete seu serviço de cliente de jogos.

Exemplo de cliente

AnyCompany Os jogos aprimoraram a segurança de seu próximo título adotando uma abordagem gerenciada de autenticação e autorização. Em vez de manter um sistema personalizado de nome de usuário e senha, eles usaram grupos de usuários do Amazon Cognito para gerenciar a inscrição e o login dos jogadores, e grupos de identidade para oferecer acesso anônimo aos jogadores que experimentavam o modo de treinamento antes de criar uma conta. Eles também implementaram uma lógica de autorização personalizada no jogo para reconhecer as funções de administrador definidas no Cognito, concedendo a esses usuários acesso a recursos especiais de gerenciamento no jogo.

Etapas de implementação

- Use grupos de usuários do Amazon Cognito para gerenciar a autenticação com tokens seguros JWTs, como habilitar recursos como cadastro, login e redefinição de senha.
- Recupere credenciais de curta duração dos grupos de identidade do Amazon Cognito para que usuários anônimos interajam com segurança com os serviços. AWS
- Implemente back-ends de jogos personalizados usando o Amazon API Gateway para uma lógica de autenticação personalizada do lado do servidor.

GAMESEC03-BP03 Use seu serviço de back-end de jogos para validar as solicitações dos jogadores para participar de um jogo multijogador

Normalmente, em jogos multijogador, um jogador entrará em uma sessão de jogo selecionando uma opção diretamente de uma lista de sessões disponíveis ou enviará uma solicitação para encontrar uma partida. A última abordagem coloca a responsabilidade do desenvolvedor do jogo de localizar uma sessão de jogo qualificada e fornecer as informações de conexão (geralmente um endereço IP e número de porta) ao cliente do jogo do jogador. A implementação pode variar dependendo do gênero do jogo que você está desenvolvendo, mas, independentemente disso, é uma prática recomendada de segurança realizar a validação do lado do servidor da solicitação de um jogador para entrar em um jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Por exemplo, em um jogo multijogador baseado em sessão, uma solicitação de um jogador para entrar em uma sessão de jogo deve ser validada pelo software do servidor de jogos com o serviço de matchmaking de back-end do jogo antes de autorizar sua conexão com o servidor. Quando um jogador solicita entrar em uma sessão de jogo, o servidor do jogo deve verificar a solicitação de um identificador exclusivo, como um ID de sessão do jogador e um tíquete gerado pelo servidor que foi fornecido anteriormente ao cliente do jogo pelo serviço de matchmaking de back-end do jogo.

Ao iniciar a conexão com o servidor do jogo, seu software do lado do servidor pode usar essas informações para verificar com o serviço de matchmaking se a solicitação de conexão do jogador é válida e verificar se o jogador não está entrando em uma vaga anteriormente reservada na sessão de jogo para outro jogador.

Para jogos hospedados na Amazon GameLift, consulte client/server Interações de jogos com GameLift servidores da Amazon para ver um exemplo de como esse tipo de validação do lado do servidor pode ser implementado.

Exemplo de cliente

Durante o lançamento beta inicial AnyCompany da Games, eles descobriram que os jogadores estavam contornando seu sistema de matchmaking conectando-se diretamente aos servidores do jogo, levando a sérios problemas de integridade competitiva. Quando jogadores de alto escalão descobriram que podiam compartilhar endereços IP do servidor com amigos, eles começaram a contornar o sistema de matchmaking baseado em habilidades, resultando em jogadores experientes participando de partidas novatas e criando uma experiência frustrante para novos jogadores. AnyCompany Os jogos responderam implementando um sistema de validação do lado do servidor que gerou tíquetes de sessão exclusivos para cada solicitação de matchmaking. O sistema exigia que o jogador e o matchmaking solicitassem tickets IDs e verificassem as tentativas de conexão com seu serviço de back-end de matchmaking.

Etapas de implementação

- Valide as solicitações de ingresso de jogadores no lado do servidor usando identificadores exclusivos, como sessão do jogador e tíquetes gerados pelo servidor. IDs
- Confirme a validade das solicitações de conexão com o serviço de matchmaking para bloquear o acesso não autorizado.

- Verifique se as vagas reservadas nas sessões de jogo não são acessadas por jogadores não autorizados durante o processo de validação.

GAMESEC03-BP04 Aplique uma política de segurança rígida para contas de usuário de jogadores, exigindo uma senha forte

Se um jogo oferecer aos jogadores a capacidade de criar uma conta de usuário com uma senha, você deve exigir que as senhas dos jogadores sigam políticas rígidas. Por exemplo, os grupos de usuários do Amazon Cognito oferecem a capacidade de definir requisitos de senha para contas de usuário. Estabelecer uma política de senha forte pode proteger as contas de seus jogadores de serem invadidas por engenharia social e ataques de força bruta.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Exemplo de cliente

AnyCompany Os jogos enfrentaram uma crise quando seu popular título sofreu uma onda de sequestros de contas devido a políticas de senha fracas. Jogadores que usavam senhas simples, como “password123", estavam se tornando vítimas de ataques automatizados de força bruta, resultando na perda de itens e no comprometimento da moeda do jogo. Para combater isso, a AnyCompany Games reformulou seu sistema de login e determinou que as senhas não fossem usadas anteriormente, incluíssem pelo menos uma letra maiúscula, um número, um caractere especial e um tamanho mínimo de 15 caracteres.

Etapas de implementação

- Exija políticas de senha fortes para contas de jogadores para aumentar a segurança.
- Use grupos de usuários do Amazon Cognito para definir e aplicar requisitos de senha.

GAMESEC03-BP05 Ofereça uma opção para os jogadores configurarem a autenticação multifator (MFA) em suas contas

As contas dos jogadores podem ser um trunfo para pessoas mal-intencionadas, principalmente em jogos que oferecem suporte a moedas e compras no jogo. Devido à generalização da invasão de contas de jogadores e dos ataques de engenharia social, ofereça aos jogadores a opção de aumentar a segurança de suas contas configurando a autenticação multifatorial (MFA).

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Quando um jogador tenta acessar sua conta usando o MFA, um código temporário é enviado para seu endereço de e-mail, número de telefone ou um aplicativo móvel de autenticação multifator desenvolvido especificamente. Para se autenticar com sucesso, o jogador deve então inserir o código no sistema de login dentro de um período de tempo limitado.

O MFA também pode ser usado para ajudar a proteger contas que estão tentando se autenticar em uma nova localização geográfica, contas que foram sinalizadas pelo suporte ao jogador por possíveis atividades maliciosas e até mesmo contas que não estão conectadas ao jogo por um longo período.

Por exemplo, grupos de usuários do Amazon Cognito podem configurar a autenticação multifatorial em diretórios de usuários.

Etapas de implementação

- Ative a autenticação multifator (MFA) para melhorar a segurança da conta do jogador.
- Use códigos temporários enviados por e-mail, telefone ou aplicativos de MFA para verificar o acesso à conta.
- Aplique o MFA para novas localizações geográficas, contas sinalizadas ou contas com inatividade prolongada.

Controle de acesso

GAMESEC04: Como você bloqueia o acesso não autorizado ao conteúdo do jogo?

Os jogos modernos incluem uma quantidade significativa de conteúdo, como conteúdo para download (DLC), que é um aspecto importante do engajamento do jogador e da monetização do jogo. Os jogadores esperam um fluxo contínuo de novos personagens, níveis e desafios, exigindo que os desenvolvedores de jogos atendam à demanda constante por novos conteúdos para reter os jogadores. A variedade e o tamanho do conteúdo podem variar muito, dependendo do tipo de jogo e do dispositivo no qual o jogo é jogado. Independentemente do sistema do jogo, proteja o conteúdo do jogo contra acesso não autorizado.

Práticas recomendadas

- GAMESEC04-BP01 Restrinja o acesso de conteúdo para download a clientes e usuários autorizados
- GAMESEC04-BP02 Limite o acesso de origem às redes autorizadas de entrega de conteúdo () CDNs
- GAMESEC04-BP03 Implemente restrições geográficas para limitar o acesso não autorizado
- GAMESEC04-BP04 Restrinja o acesso ao conteúdo com soluções de gerenciamento de direitos digitais (DRM)

GAMESEC04-BP01 Restrinja o acesso de conteúdo para download a clientes e usuários autorizados

Restrinja o acesso ao conteúdo do jogo por aplicativos e clientes autorizados. Considere usar o Amazon S3 como uma origem econômica e escalável para armazenar conteúdo de jogos para download e o Amazon para fornecer entrega de conteúdo com desempenho global CloudFront aos jogadores. Ambos os serviços fornecem mecanismos integrados para restringir o acesso aos dados armazenados, como restringir o acesso a usuários autenticados.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Conceder acesso ao conteúdo armazenado no Amazon S3

Quando você precisa conceder acesso ao conteúdo armazenado no S3, há vários fatores a serem considerados. Por padrão, somente quem Conta da AWS criou um bucket do S3 pode acessar os objetos armazenados nele. Para conceder acesso aos seus aplicativos internos e gerenciar o conteúdo armazenado nos buckets do Amazon S3, use AWS Identity and Access Management (IAM) para criar políticas que forneçam acesso adequado.

As funções do IAM podem ser associadas a usuários, sistemas ou aplicativos federados hospedados em serviços, como Amazon EC2 AWS Lambda, e aplicativos baseados em contêineres hospedados no Amazon EKS e no Amazon ECS. Por exemplo, você pode usar o AWS SDK ou AWS CLI publicar e gerenciar ativos de conteúdo de jogos em buckets do S3. Para dar suporte a esse caso de uso, você pode criar uma função do IAM com acesso adequado para ler e gravar conteúdo do jogo em seus buckets do S3 e associá-la às EC2 instâncias que hospedam seu software e scripts.

Políticas baseadas em recursos podem ser definidas para seu bucket e para objetos específicos. As políticas de bucket do S3 estão associadas a um bucket do S3 e podem ser usadas para restringir o acesso ao bucket e aos objetos dentro dele, bem como conceder acesso aos recursos do Amazon S3 a partir de outras contas. Por exemplo, em cenários em que várias equipes ou estúdios de desenvolvimento de jogos separados estão trabalhando no mesmo conteúdo do jogo e exigem o mesmo acesso ao conteúdo hospedado centralmente no Amazon S3, você pode usar uma política de bucket do S3 para definir permissões para acesso entre contas aos recursos do S3. Considere o uso de pontos de acesso S3, que podem simplificar o gerenciamento do acesso aos dados compartilhados criando pontos de acesso com nomes e permissões específicos para cada aplicativo ou conjunto de aplicativos. A documentação do Amazon S3 contém melhores práticas adicionais para controle de acesso no Amazon S3.

Granting short-term access to your content

Quando o acesso for necessário apenas por um tempo limitado específico, gere um temporário URLs que conceda acesso de curto prazo ao seu conteúdo. O Amazon S3 fornece suporte para geração de objetos pré-assinados URLs, o que permite que proprietários de objetos concedam acesso por tempo limitado a objetos no Amazon S3 sem atualizar sua política de bucket. Ao fazer isso, o usuário final ou o aplicativo que está recebendo acesso não precisa ter uma conta ou permissões do IAM e, em vez disso, usa o URL pré-assinado para acessar o conteúdo.

Essa é uma prática recomendada comumente usada em vários casos de uso de jogos, como conceder aos jogadores autorizados acesso ao conteúdo para download ao qual eles têm direito e fornecer acesso temporário ao conteúdo do jogo por tempo limitado. O Presigned também URLs pode ser usado para fornecer permissões temporárias para carregar conteúdo em um bucket do S3. Por exemplo, você pode considerar usar uma URL pré-assinada para fornecer a um jogador acesso ao upload de registros do cliente para ajudar sua equipe de suporte a solucionar um caso de suporte para jogadores.

Usando uma rede de distribuição de conteúdo para fornecer acesso ao seu conteúdo

Embora seus aplicativos, desenvolvedores de jogos, artistas e outros funcionários precisem de acesso direto ao conteúdo nos buckets do S3 para fins de desenvolvimento e gerenciamento, use uma rede de distribuição de conteúdo para fornecer acesso ao conteúdo que está disponível publicamente para jogadores ou outros usuários pela Internet. Essa abordagem melhora o desempenho do download e reduz os custos ao armazenar em cache o conteúdo acessado com frequência. A Amazon CloudFront pode distribuir globalmente seu conteúdo armazenando em cache

e entregando-o mais perto de seus jogadores, ao mesmo tempo em que reduz a carga na origem do download do seu jogo, como o Amazon S3.

Em vez de veicular seu conteúdo público diretamente dos buckets do S3, é recomendável manter esse conteúdo privado e disponibilizá-lo publicamente usando. CloudFront CloudFront pode ser configurado para exigir que os jogadores acessem seu conteúdo privado (como o download de um novo jogo somente para jogadores pagos) usando cookies assinados URLs ou assinados. Em seguida, você pode desenvolver seu aplicativo para criar e distribuir assinados URLs para usuários autenticados ou para enviar cabeçalhos de configuração de cookies que definem cookies assinados para usuários autenticados. Ao criar cookies assinados URLs ou assinados para controlar o acesso aos seus arquivos, você pode especificar uma data e hora de término, após as quais o URL e os cookies não são mais válidos.

Opcionalmente, você também pode especificar o endereço IP ou o intervalo de endereços dos computadores que podem ser usados para acessar seu conteúdo, o que é útil se você quiser restringir o acesso a parceiros de estúdios de desenvolvimento de jogos específicos ou redes de prestadores de serviços. Use cookies assinados quando quiser fornecer acesso a vários arquivos restritos ou se não quiser alterar seus arquivos atuais URLs. Use assinado URLs quando quiser restringir o acesso a arquivos individuais ou se seus usuários estiverem usando um cliente que não suporta cookies. Os cookies assinados têm URLs precedência sobre os cookies assinados.

Etapas de implementação

- Use funções do IAM e políticas de bucket para conceder acesso adequado aos buckets do S3 para aplicativos internos, equipes ou cenários entre contas.
- Gere conteúdo pré-assinado URLs para conceder acesso de curto prazo a objetos do S3, adequado para conteúdo para download ou uploads temporários, como registros de clientes.
- Use a Amazon CloudFront com URLs assinaturas ou cookies para fornecer conteúdo privado com mais segurança a usuários autenticados

GAMESEC04-BP02 Limite o acesso de origem às redes autorizadas de entrega de conteúdo () CDNs

Impeça que os usuários burlem suas redes de distribuição de conteúdo para acessar diretamente o conteúdo de sua origem, como seus buckets do Amazon S3. É importante restringir o acesso à sua origem somente aos autorizados CDNs, o que reduz os custos de transferência de dados decorrentes da veiculação desnecessária de conteúdo fora da origem. Também melhora sua postura

de segurança ao direcionar o acesso público ao seu conteúdo de origem pelo mesmo ponto de entrada, onde você pode implantar controles de segurança de ponta, como filtragem de AWS WAF camada 7, injeção e inspeção de parâmetros de solicitação HTTP relacionados à segurança e proteções distribuídas de negação de serviço (S). DDo

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Para implementar esses controles para uma origem do Amazon S3, você pode usar uma identidade de acesso de CloudFront origem da Amazon (OAI), que verifica se as solicitações para seus objetos do S3 são originárias da sua distribuição. CloudFront Associe-se AWS WAF à sua CloudFront distribuição para fornecer filtragem de camada 7. No entanto, se você estiver veiculando conteúdo adicional CDNs, poderá configurar a CDN para inserir um ou mais cabeçalhos HTTP personalizados nas solicitações de origem, que podem ser inspecionadas AWS WAF para verificar se o tráfego de entrada foi originado do seu provedor de CDN autorizado.

Essa abordagem também é útil para ajudar a evitar que os usuários burlem seus provedores de CDN quando sua origem está hospedada por trás de um Application Load Balancer (ALB). ALBs pode ser associado a quatro AWS WAF proteções de camada 7. Você pode configurar AWS WAF para inserir um cabeçalho HTTP personalizado que será inspecionado pelo seu ALB para processar e inspecionar o tráfego de entrada para o balanceador de carga. AWS WAF

Exemplo de cliente

AnyCompany A Games implementa restrições de acesso ao Origin para ajudar a proteger seus ativos de jogos, conteúdo para download e arquivos de patch contra acesso direto não autorizado, o que pode permitir que os jogadores ignorem as verificações de segurança ou obtenham conteúdo premium sem a autenticação adequada. Essa abordagem permite que eles monitorem os padrões de acesso ao conteúdo por meio de um ponto centralizado, facilitando a identificação de comportamentos suspeitos de download que possam indicar a presença de ataques coordenados ou redistribuição não autorizada de conteúdo.

Etapas de implementação

- Use a identidade de acesso de CloudFront origem da Amazon (OAI) para restringir o acesso direto aos objetos do S3
- AWS WAF Associe-se ao CloudFront nosso ALB para fornecer filtragem de camada 7 e ajudar a proteger contra ataques DDo S e solicitações maliciosas.

- Configure cabeçalhos HTTP personalizados no Cloudfront para verificar se o tráfego de entrada vem de fontes autorizadas.

GAMESEC04-BP03 Implemente restrições geográficas para limitar o acesso não autorizado

Quando um jogador solicita seu conteúdo, a CloudFront Amazon fornece o conteúdo solicitado no ponto de presença mais próximo, independentemente de onde o player esteja localizado. No entanto, pode haver cenários em que você precise restringir a forma como seu conteúdo é acessível por usuários em partes específicas do mundo. Por exemplo, você pode ter uma estratégia de implantação de jogos contínuos que libera conteúdo em fases de country-by-country forma contínua, ou talvez precise seguir os controles de acesso específicos de cada país.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Você pode usar restrições geográficas, também conhecidas como bloqueio geográfico, para impedir que jogadores em localizações geográficas específicas acessem o conteúdo que você está distribuindo por meio de uma CloudFront distribuição. Esse recurso permite restringir o acesso aos arquivos associados a uma distribuição e restringir o acesso no nível do país. Como alternativa, você pode usar um serviço de geolocalização terceirizado para restringir o acesso a um subconjunto dos arquivos associados a uma distribuição ou para restringir o acesso em uma granularidade mais fina do que no nível do país.

Ao usar restrições CloudFront geográficas, você pode permitir que seus jogadores acessem seu conteúdo somente se estiverem em um dos países que estão em uma lista de países aprovados. Você também pode impedir que seus jogadores acessem seu conteúdo se eles estiverem em um dos países que estão na lista negada de países banidos. Se uma solicitação for recebida de uma localização geográfica bloqueada, CloudFront retornará um código de status HTTP 403 Forbidden para o player. É importante observar que isso funciona bem para conteúdo não confidencial e não deve ser usado como proteção independente para PII ou artefatos confidenciais de jogos.

Etapas de implementação

- Use restrições CloudFront geográficas para permitir ou negar o acesso ao conteúdo com base nas listas de permissão ou negação em nível de país.

- Retorne um código de status HTTP 403 Forbidden para solicitações originadas de localizações geográficas bloqueadas.
- Evite confiar apenas em restrições geográficas para proteger conteúdo confidencial ou PII

GAMESEC04-BP04 Restrinja o acesso ao conteúdo com soluções de gerenciamento de direitos digitais (DRM)

Considere restringir o acesso ao conteúdo do seu jogo usando ferramentas de criptografia robustas, como uma solução de gerenciamento de direitos digitais (DRM). Esse tipo de solução pode ser usado para criptografar seu conteúdo privado e distribuir as chaves de decodificação para jogadores autorizados.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

As soluções de DRM são recomendadas em situações em que você deseja permitir que os jogadores baixem o conteúdo do jogo mais cedo, mas não quer que eles possam acessar ou reproduzir o conteúdo até um horário predeterminado. Por exemplo, isso é comum em situações em que os jogadores podem pré-encomendar um jogo e configurar seu cliente de jogo para começar automaticamente a baixar os arquivos criptografados mais cedo. Essa estratégia verifica se o jogo foi baixado e está pronto para ser jogado assim que o jogo for lançado oficialmente. Depois que o jogo for lançado, o cliente do jogo do jogador poderá solicitar chaves de decodificação da solução de back-end DRM para que ele possa descriptografar os arquivos baixados anteriormente e começar a jogar.

Os sistemas DRM também são usados para bloquear a redistribuição e manipulação não autorizadas de jogos depois de terem sido baixados e instalados por um jogador autorizado. Os sistemas DRM exigem integração com a origem para trocar chaves de criptografia e autorizar os jogadores a recuperar a chave de decodificação. Os fornecedores comerciais de DRM oferecem uma variedade de soluções com recursos e suporte para diferentes dispositivos.

Etapas de implementação

- Use soluções de DRM para criptografar conteúdo privado de jogos e distribuir chaves de decodificação para jogadores autorizados.
- Ative o pré-download de arquivos criptografados para jogos pré-encomendados, desbloqueando o acesso com chaves de decodificação no momento do lançamento.

- Integre sistemas DRM com a origem para gerenciar chaves de criptografia e bloquear a redistribuição ou manipulação não autorizada de conteúdo.

Detecção

GAMESEC05: Como você monitora e analisa o comportamento de uso do jogador em seu jogo?

Monitorar e analisar o comportamento de uso do jogador é essencial para os estúdios de jogos, pois permite detectar ameaças à segurança, trapaças e outras formas de comportamento abusivo que podem comprometer a integridade do jogo e a segurança do jogador. Ao rastrear padrões como taxas de progressão incomuns, transações anormais no jogo ou comportamentos de comunicação suspeitos, você pode identificar possíveis trapaceiros, contas fraudulentas ou ameaças coordenadas antes que elas afetem significativamente a experiência do jogador.

Práticas recomendadas

- GAMESEC05-BP01 Implemente uma estratégia abrangente de coleta de dados para monitorar o comportamento do jogador
- GAMESEC05-BP02 Colete, armazene e analise os registros de uso do jogador para detectar comportamentos impróprios

GAMESEC05-BP01 Implemente uma estratégia abrangente de coleta de dados para monitorar o comportamento do jogador

Para manter uma experiência positiva para o jogador, implemente uma estratégia abrangente de coleta e análise de dados. Capturar, armazenar e analisar dados relevantes fornece informações sobre como os jogadores interagem com os recursos do seu jogo e entre si. Essa abordagem baseada em dados pode orientar a tomada de decisões, aprimorar o engajamento e a retenção dos jogadores, otimizar as estratégias de monetização e, por fim, melhorar a experiência geral do jogador.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Implemente sistemas de coleta de dados para capturar e registrar ações relevantes do jogador, como sessões de jogo, progresso, conquistas, compras, interações com elementos do jogo e atividades sociais. Colete dados do lado do servidor, como carga do servidor, tráfego de rede e registros de erros, para monitorar o desempenho técnico e identificar possíveis problemas. Obtenha feedback dos jogadores por meio de pesquisas, fóruns, tickets de suporte e canais de mídia social para entender suas experiências e preferências.

Ao armazenar os dados do jogo, estabeleça um data warehouse ou data lake centralizado para armazenar e organizar os dados coletados e implemente pipelines para limpeza, transformação e agregação de dados para preparar os dados para uma análise eficiente.

Depois de armazenar os dados, analise-os para obter informações como retenção e rotatividade de jogadores, estratégias de monetização e uso de recursos por meio de ferramentas de visualização de dados.

Etapas de implementação

- Capture e registre ações do jogador, métricas do servidor e feedback para monitorar as interações e o desempenho técnico.
- Use um data warehouse centralizado, como o Amazon Redshift ou o data lake S3, para armazenar, limpar, transformar e organizar dados de jogos para análise.
- Analise os dados coletados com ferramentas de visualização, como o Amazon Quicksight, para obter informações sobre retenção de jogadores, monetização e uso de recursos.

GAMESEC05-BP02 Colete, armazene e analise os registros de uso do jogador para detectar comportamentos impróprios

Instrumente seu jogo para coletar registros para entender como os jogadores usam os recursos do seu jogo e como eles interagem com outros jogadores. Você pode então bloquear atividades não autorizadas que podem degradar a experiência do jogador.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Envie eventos de log estruturados para o Game Analytics Pipeline usando uma solução de registro, como Amazon CloudWatch Logs ou Amazon OpenSearch Service, ou por meio de uma solução

de um AWS parceiro, como Datadog, Sumo Logic, New Relic, Honeycomb.io ou Splunk. Estruture esses registros de uso do jogador para que eles possam ser usados para detectar quando ações específicas dos jogadores precisam ser investigadas.

Depois de capturar os dados, considere a implementação de ferramentas para detectar comportamentos de uso inadequados. Por exemplo, se seu jogo tiver recursos sociais, como mensagens de jogadores, bate-papo por voz ou fóruns on-line, salve os registros desses engajamentos de jogadores em um formato que possa ser analisado para fins de moderação.

Configure o recurso de bate-papo por voz do seu jogo para exportar gravações para o Amazon S3 e use o Amazon Transcribe para converter a fala de áudio em formato de texto, que pode ser armazenado para processamento. Como alternativa, você pode realizar a transcrição de streaming em tempo real integrando o serviço de bate-papo por voz de back-end do jogo diretamente com a API Transcribe para transcrever o streaming de áudio em tempo real. As equipes de moderação podem revisar manualmente o conteúdo e, quando o conteúdo estiver em um formato padrão, você também poderá usar os serviços de AWS IA/ML para realizar a moderação automaticamente. O Amazon Comprehend pode ser usado para realizar processamento de linguagem natural (NLP) para descobrir informações do texto não estruturado, o que pode classificar e organizar as conversas em tópicos relevantes e identificar comportamentos impróprios, como palavrões.

Etapas de implementação

- Colete, armazene e analise os registros de uso do jogador.
- Use AWS serviços de inteligência artificial e aprendizado de máquina para analisar e obter informações mais eficientes sobre seus registros de uso de jogadores.

Proteção da infraestrutura

Consulte o whitepaper do Well-Architected Framework para obter as melhores práticas de proteção de infraestrutura para segurança que se aplicam às cargas de trabalho de jogos.

GAMESEC06: Como você monitora e responde às ameaças à infraestrutura?

Monitorar e responder às ameaças à infraestrutura é fundamental para os estúdios de jogos porque sua infraestrutura representa a espinha dorsal que suporta milhões de jogadores simultâneos, processa transações em dinheiro real e armazena dados valiosos dos jogadores e conteúdo

proprietário do jogo. É essencial que os estúdios de jogos implementem sistemas de monitoramento e processos de resposta a incidentes que possam preservar a integridade das experiências dos jogadores e das operações comerciais.

Práticas recomendadas

- GAMESEC06-BP01 Use ferramentas para detectar e responder às ameaças à sua infraestrutura
- GAMESEC06-BP02 Use ferramentas de inteligência artificial e aprendizado de máquina para automatizar aspectos de sua estratégia de proteção de infraestrutura
- GAMESEC06-BP03 Use insights de registros em nível de sistema para melhorar continuamente sua estratégia de proteção de infraestrutura

GAMESEC06-BP01 Use ferramentas para detectar e responder às ameaças à sua infraestrutura

Para monitorar continuamente atividades maliciosas e comportamentos não autorizados em seu AWS ambiente, considere usar a Amazon GuardDuty. GuardDuty identifica ameaças monitorando o comportamento da conta, a atividade da rede e os padrões de acesso aos dados em seu ambiente. Ele analisa eventos em várias fontes de dados, como registros de CloudTrail eventos, registros de fluxo do Amazon VPC e registros de DNS para detectar possíveis ameaças. Com a integração com o Amazon CloudWatch Events e o Lambda GuardDuty , os alertas podem ser encaminhados automaticamente às equipes de segurança relevantes para análise posterior.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

AWS Security Hub CSPMfornece uma visão abrangente do seu estado de segurança AWS e verifica seu ambiente de acordo com os padrões e as melhores práticas do setor de segurança. O Security Hub CSPM coleta dados de segurança de vários Contas da AWS serviços e produtos de parceiros terceirizados compatíveis, analisa suas tendências de segurança e identifica os problemas de segurança de maior prioridade. A GuardDuty integração da Amazon com o Security Hub CSPM permite que você envie descobertas para o Security Hub CSPM. GuardDuty O Security Hub CSPM pode então incluir essas descobertas em sua análise de sua postura de segurança.

É comum que pessoas mal-intencionadas usem bots para assumir contas e trapacear em jogos. O WAF Bot Control oferece visibilidade e controle sobre o tráfego comum e generalizado de bots que

pode consumir recursos em excesso, distorcer métricas, causar tempo de inatividade ou realizar outras atividades indesejadas.

Ransomware é um código malicioso projetado para obter acesso não autorizado a sistemas e conjuntos de dados e criptografar esses dados para bloquear o acesso de jogadores legítimos. Depois que o ransomware bloqueou os jogadores de seus sistemas e criptografou seus dados confidenciais, os cibercriminosos exigem um resgate antes de fornecer uma chave de decodificação para desbloquear os dados. As organizações podem ser completamente fechadas por um evento malicioso, incorrendo em custos significativos e perda de produtividade empresarial. Consulte Protegendo seu Nuvem AWS ambiente contra ransomware para ver as melhores práticas que você pode aplicar para fortalecer sua capacidade de combater o ransomware antes, durante e depois da ocorrência de um incidente.

Seu jogo pode permitir que os jogadores entrem em contato com agentes de suporte por meio de uma central de atendimento, como o Amazon Connect, ou bots de bate-papo usando o Amazon Lex. O Amazon Connect fornece suporte para monitorar conversas ao vivo e gravadas. Para analisar as interações entre jogadores e bots de bate-papo de suporte a jogadores criados com o Amazon Lex, você pode armazenar os registros de conversas dessas interações no Amazon CloudWatch Logs, que podem ser exportados para o Amazon S3 e analisados conforme descrito anteriormente.

Por fim, realize exercícios de teste de penetração como parte de sua estratégia de proteção de infraestrutura. Se você estiver realizando essas avaliações internamente ou por meio de um AWS parceiro, siga as políticas de suporte ao AWS cliente para testes de penetração.

Etapas de implementação

- Use GuardDuty a Amazon para monitorar o comportamento da conta, a atividade da rede e os padrões de acesso aos dados em busca de ameaças e integre-se ao Security Hub CSPM para obter uma visão de segurança unificada.
- Implemente o AWS WAF Bot Control para ajudar a detectar e mitigar o tráfego de bots que pode prejudicar os recursos e as experiências dos jogadores.
- Realize exercícios de testes de penetração regularmente, seguindo as políticas de suporte ao AWS cliente, para avaliar e fortalecer sua postura de segurança.

GAMESEC06-BP02 Use ferramentas de inteligência artificial e aprendizado de máquina para automatizar aspectos de sua estratégia de proteção de infraestrutura

O Amazon Lookout for Metrics usa aprendizado de máquina para detectar e diagnosticar automaticamente anomalias em seus dados comerciais e operacionais e monitora as métricas mais importantes para seus negócios com maior velocidade e precisão. O serviço também facilita o diagnóstico da causa raiz das anomalias, como uma queda repentina na receita, nos logins, nas transações ou na retenção. Ele não exige que os desenvolvedores de jogos tenham experiência em ML para configurar e podem se conectar a fontes de dados populares, incluindo Amazon S3, Amazon, CloudWatch Amazon RDS, Amazon Redshift, bem como muitos aplicativos SaaS. Por exemplo, você pode integrar o Amazon Lookout for Metrics com o Game Analytics Pipeline e outras fontes de dados para começar a analisar o comportamento e detectar anomalias.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Como alternativa, você pode optar por criar, treinar e hospedar um modelo personalizado de aprendizado de máquina usando a SageMaker IA da Amazon para tratar de casos de uso como moderação de conteúdo, detecção de toxicidade, detecção de fraudes, detecção de fraudes e muito mais.

Exemplo de cliente

AnyCompany A Games usa o Amazon Lookout for Metrics para detectar automaticamente padrões incomuns no desempenho do servidor, nas tentativas de login de jogadores ou nos volumes de transações que podem indicar ameaças de agentes mal-intencionados. Além disso, eles usaram a Amazon SageMaker AI para desenvolver modelos personalizados de aprendizado de máquina que analisam continuamente os padrões de tráfego de rede e o comportamento dos jogadores para ajudar a identificar ameaças coordenadas, como redes de bots que estão tentando explorar sua economia virtual.

Essa abordagem automatizada permite que sua equipe de segurança se concentre em investigar e responder a ameaças genuínas, em vez de monitorar manualmente milhares de métricas, ao mesmo tempo em que garante que os padrões de ameaças emergentes sejam detectados e resolvidos antes que possam afetar significativamente a disponibilidade do jogo ou a segurança do jogador.

Etapas de implementação

- Use o Amazon Lookout for Metrics Lookout for Metrics para ajudar a detectar e diagnosticar automaticamente anomalias nos principais dados comerciais e operacionais
- Integre o Amazon Lookout for Metrics com fontes de dados como o Game Analytics Pipeline, o Amazon S3 CloudWatch , ou monitore métricas como receita, logins e retenção.
- Use a Amazon SageMaker AI para criar, treinar e hospedar modelos personalizados de aprendizado de máquina para casos de uso avançados, como detecção de fraudes, prevenção de fraudes e moderação de conteúdo.

GAMESEC06-BP03 Use insights de registros em nível de sistema para melhorar continuamente sua estratégia de proteção de infraestrutura

Capture e armazene registros em nível de sistema de serviços relevantes, como registros de acesso ao servidor S3, registros de acesso e registros de CloudFront acesso do ALB. Esses registros podem ser armazenados em um bucket do S3 em sua conta e são úteis para associar as informações de uso do jogador de dentro do jogo a informações no nível do sistema, incluindo detalhes de conexão, como endereços IP, cabeçalhos de solicitação e manipulação e filtragem de solicitações relevantes que você possa ter configurado no back-end do jogo. Você pode enviar esses registros para as mesmas soluções de registro mencionadas anteriormente e analisá-los usando consultas SQL com o Amazon Athena sem exigir que os registros sejam movidos do Amazon S3.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

O Access Analyzer for S3 é um recurso que monitora suas políticas de acesso ao bucket, garantindo que as políticas forneçam somente o acesso pretendido aos seus recursos do Amazon S3. O Access Analyzer for S3 avalia suas políticas de acesso ao bucket e permite que você descubra e corrija rapidamente os buckets com acesso potencialmente não intencional.

Etapas de implementação

- Use AWS serviços de detecção de ameaças e resposta a incidentes para automatizar aspectos de sua estratégia de proteção de infraestrutura.
- Obtenha insights sobre a proteção de sua infraestrutura por meio de registros e AWS serviços em nível de sistema para inteligência artificial e aprendizado de máquina.

Proteção de dados

Ao desenvolver e arquitetar seu jogo, considere o tipo de dados que seu estúdio está coletando e como você decidiu abordar a proteção deles. Os tópicos a serem explorados nesse aspecto da segurança incluem:

- Como você escolheu identificar e classificar seus dados
- Como você está protegendo os dados em repouso
- Como você está protegendo os dados em trânsito

Não há práticas recomendadas de proteção de dados específicas para o Games Lens. Consulte o whitepaper do Well-Architected Framework para obter as melhores práticas em proteção de dados para segurança.

Resposta a incidentes

GAMESEC07: Como você está definindo e aplicando políticas para responder à má conduta e ao comportamento abusivo dos jogadores?

A má conduta e os comportamentos abusivos dos jogadores podem afetar significativamente a experiência de seus jogadores. O mau comportamento do jogador pode afastar jogadores legítimos, levando à diminuição da retenção de jogadores, redução da receita de compras no jogo e avaliações negativas que podem prejudicar a reputação e as vendas futuras de um jogo.

Defina políticas que promovam ações positivas entre seus jogadores e determine como você aplicará essas políticas.

Práticas recomendadas

- GAMESEC07-BP01 Implemente um plano de resposta a incidentes para lidar com malfeitores e comportamento abusivo
- GAMESEC07-BP02 Banir contas associadas a malfeitores

GAMESEC07-BP01 Implemente um plano de resposta a incidentes para lidar com malfeitores e comportamento abusivo

Crie um plano de ação para responder aos malfeitores e ao comportamento abusivo em seu jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Considere fatores como quando suspender temporariamente ou banir permanentemente jogadores e por quanto tempo desativar as credenciais de jogadores temporariamente suspensos.

Exemplo de cliente

AnyCompany A Games cria um sistema hierárquico de resposta a incidentes no qual infrações menores, como mensagens de bate-papo inapropriadas, resultam em suspensões automáticas da conta por 24 horas, enquanto violações mais graves, como trapaça ou assédio, desencadeiam suspensões imediatas de 7 dias com revisão obrigatória por moderadores humanos.

Além disso, a AnyCompany Games estabelece procedimentos de escalonamento nos quais infratores reincidentes enfrentam suspensões progressivamente mais longas. Eles criam processos de apelação que permitem que jogadores falsamente sinalizados contestem ações automatizadas, mantendo a segurança por meio de requisitos de verificação de identidade.

GAMESEC07-BP02 Banir contas associadas a malfeitores

Se não forem mitigados, os comportamentos abusivos em um jogo podem continuar a ter um impacto negativo na experiência de jogo de outras pessoas e devem ser mitigados o mais rápido possível. Implemente um processo para impor proibições ou outras formas de restrições a agentes mal-intencionados que comprovadamente violaram seus termos de serviço.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Normalmente, as regras e o processo de avaliação para determinar as circunstâncias da imposição desses tipos de restrições serão determinados por funcionários, como uma equipe de comunidade de jogadores ou uma equipe de confiança e segurança de sua organização. Depois de identificar os agentes mal-intencionados, execute um fluxo de trabalho predeterminado para agir sobre os atores identificados.

Por exemplo, AWS Step FunctionsAWS Lambdafunções podem ser usadas para executar um fluxo de trabalho automatizado que aceita um lote de contas de jogadores como entrada. O fluxo de trabalho então atualiza as entradas em uma tabela do Amazon DynamoDB chamada Bans, que pode incluir detalhes sobre a conta do jogador, o motivo do banimento e a duração.

Dependendo do design do seu sistema de gerenciamento de jogos e contas e do tipo de abuso que você enfrenta por parte de pessoas mal-intencionadas, mantenha um sistema de banimento de registros separado do seu sistema de gerenciamento de contas. Talvez você não queira desativar a conta do jogador do seu sistema de gerenciamento de contas, optando por simplesmente desativar a capacidade dele de jogar seu jogo. Isso pode ser útil em situações em que as credenciais da conta do jogador são usadas para acessar vários jogos com termos de serviço ou políticas diferentes.

Etapas de implementação

- Defina e aplique políticas para responder a comportamentos abusivos de pessoas mal-intencionadas.
- Use AWS serviços para automatizar suas respostas a pessoas mal-intencionadas.

Recursos

- AWS Guia técnico de resposta a incidentes de segurança
- AWS Blog de Machine Learning: detecte usuários reais e ativos e detenha pessoas mal-intencionadas usando o Amazon Rekognition Face Liveness
- AWS Soluções para jogos: Community Health

Segurança de aplicações

GAMESEC08: Como você protege seu pipeline? CI/CD

Um CI/CD pipeline de desenvolvimento de jogos geralmente é composto por servidores e armazenamento de controle de origem altamente disponíveis, recursos computacionais para executar suas compilações e software para realizar testes automatizados, além da conectividade de rede adequada de suas máquinas de desenvolvimento. Proteger seu CI/CD pipeline é importante para proteger informações confidenciais, preservar a integridade do código e manter versões

confiáveis. A incorporação de governança e proteções permite a agilidade do desenvolvedor e, ao mesmo tempo, mantém as boas práticas de segurança.

Como os jogos geralmente lidam com o processamento de pagamentos, armazenam informações pessoais e mantêm economias virtuais que valem dinheiro real, uma violação de segurança no processo de desenvolvimento pode resultar em perdas financeiras significativas, penalidades regulatórias e perda da confiança dos jogadores.

Ao integrar as proteções, as organizações mantêm a visibilidade e o controle sobre o processo de entrega de software, permitindo uma resposta rápida a incidentes e promovendo uma cultura de práticas seguras de codificação.

Práticas recomendadas

- GAMESEC08-BP01 Aplique segurança em todas as etapas do pipeline de CI/CD

GAMESEC08-BP01 Aplique segurança em todas as etapas do pipeline de CI/CD

Proteções como controles de acesso, separação de tarefas e trilhas de auditoria fornecem proteção contra acesso não autorizado ou atividades maliciosas.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Seu pessoal, seus processos e sua tecnologia também devem proteger o pipeline. As pessoas mais próximas do código devem estabelecer práticas de codificação seguras e garantir que as sigam. Repita seus processos continuamente para verificar se há consistência no nível de segurança em todo o pipeline. Por fim, implemente a tecnologia para verificar se as melhores práticas e processos não são ignorados.

Exemplo de cliente

AnyCompany Os jogos implementam controles de acesso baseados em funções, nos quais somente desenvolvedores seniores podem aprovar alterações no código do sistema anti-trapaça, ao mesmo tempo em que exigem revisões obrigatórias do código dos membros da equipe de segurança para componentes que lidam com os dados de pagamento dos jogadores.

Seu CI/CD pipeline executa automaticamente verificações de validação do modelo de ameaças, garantindo que novos recursos, como um mercado de negociação de jogadores, sejam testados em

relação a vetores de ataque previamente identificados, como explorações de duplicação de itens ou tentativas fraudulentas de transação.

Etapas de implementação

- Forneça permissões aos usuários com base no princípio do menor privilégio.
- Use AWS CloudTrail para auditar chamadas de API feitas nos serviços usados no pipeline.
- Use ganchos de pré-confirmação para verificar se o código está seguindo as práticas gerais e as políticas da empresa.

Automatize a segurança

GAMESEC09: Como você automatiza a segurança em seu pipeline? CI/CD

Integre medidas de segurança em seu CI/CD pipeline para manter uma postura de segurança robusta durante todo o ciclo de vida do desenvolvimento. Esse processo oferece muitos dos mesmos benefícios de ter a segurança do seu pipeline. Ter um CI/CD pipeline seguro reduz a probabilidade de eventos de segurança que poderiam atrasar o cronograma de desenvolvimento do jogo.

Fornecer segurança em seu CI/CD pipeline envolve a implementação das melhores práticas e ferramentas de segurança em cada estágio do ciclo de desenvolvimento. Ter a segurança em andamento também permite reduzir o tempo das análises de segurança.

Automatizar a segurança em seu CI/CD pipeline é particularmente crucial para verificar se os controles de segurança são implementados e testados de forma consistente com cada alteração de código. A implementação das ferramentas e automação adequadas pode oferecer jogos seguros e protegidos.

Práticas recomendadas

- GAMESEC09-BP01 Integre ferramentas e automação para reduzir o tempo médio de análises de segurança

GAMESEC09-BP01 Integre ferramentas e automação para reduzir o tempo médio de análises de segurança

Para identificar vulnerabilidades de segurança, as organizações podem usar uma variedade de ferramentas e serviços diferentes, como Static Application Security Testing (SAST) e Dynamic Application Security Testing (DAST). O SAST é uma forma de revisar o código-fonte e determinar a vulnerabilidade de segurança. O DAST é uma forma de caixa preta de testar seu código que testa seus aplicativos sem examinar o código-fonte.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Outra ferramenta que as organizações podem usar é a Análise de Composição de Software (SCA), que avalia a segurança de suas dependências de terceiros ou de código aberto. Para uma abordagem mais manual, revisões seguras de código podem ser implementadas em todo o pipeline.

Exemplo de cliente

AnyCompany Os jogos usam ferramentas SAST para sinalizar automaticamente possíveis falhas de segurança durante o processo de desenvolvimento. Eles também usam ferramentas DAST para simular ameaças contra a execução de compilações de jogos para validar se os controles de segurança estão funcionando conforme o esperado. Além disso, a AnyCompany Games integra ferramentas de verificação de dependências em seu processo de desenvolvimento para identificar automaticamente vulnerabilidades conhecidas em bibliotecas e mecanismos de jogos de terceiros.

Etapas de implementação

- Use a Amazon CodeGuru como uma ferramenta SAST.
- Use ferramentas de código aberto como o OWASP Dependency Check,, ou. SonarQube OWASPZap

Recursos

- Segurança para desenvolvedores

Modelagem de ameaças

GAMESEC10: Como você integra a modelagem de ameaças ao ciclo de vida de desenvolvimento de aplicativos da sua organização?

A modelagem de ameaças é o processo de identificar e priorizar possíveis ameaças ao seu aplicativo e determinar as soluções que podem ser usadas para mitigá-las. Essa prática se tornou cada vez mais importante à medida que os jogos evoluíram para sistemas complexos e conectados que lidam com dados confidenciais do usuário e transações em dinheiro real.

Integre a modelagem de ameaças como um exercício contínuo para apoiar a segurança do jogo, não apenas na fase inicial de design, mas à medida que o jogo continua crescendo e evoluindo.

Práticas recomendadas

- GAMESEC10-BP01 Determine quando e como concluir os exercícios de modelagem de ameaças em todo o ciclo de vida de desenvolvimento de aplicativos

GAMESEC10-BP01 Determine quando e como concluir os exercícios de modelagem de ameaças em todo o ciclo de vida de desenvolvimento de aplicativos

Não há uma única maneira melhor de abordar a modelagem de ameaças. Os detalhes de quando e como fazer isso variam de acordo com as necessidades exclusivas do seu estúdio de jogos. Por exemplo, dependendo do tamanho do seu estúdio, você pode ter membros da equipe envolvidos em um ou vários aspectos do processo de modelagem de ameaças.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

O blog AWS de segurança fornece uma visão geral das considerações que você deve ter em mente ao elaborar sua estratégia para modelagem de ameaças, como:

- Quais membros e personas da sua equipe devem estar envolvidos na modelagem de ameaças
- Como determinar as ferramentas de fluxo de trabalho apropriadas a serem usadas

- Como determinar a propriedade de vários aspectos da modelagem de ameaças
- Como identificar e avaliar os controles de segurança a serem usados em seu design de carga de trabalho

Exemplo de cliente

AnyCompany Os jogos começam catalogando ativos valiosos, como dados do jogador, código e algoritmos do jogo, moedas do jogo, conteúdo gerado pelo usuário e propriedade intelectual, como conteúdo inédito ou mecanismos proprietários. Eles consideram diferentes tipos de malfeitores em potencial, como trapaceiros que buscam vantagens injustas, malfeitores tentando roubar dados pessoais ou financeiros e usuários mal-intencionados tentando atrapalhar o jogo.

Durante todo o processo de desenvolvimento, a AnyCompany Games usa modelos de ameaças para orientar práticas seguras de codificação e influenciar as estratégias de teste para se concentrar em áreas de alto risco. Antes do lançamento do jogo, eles conduzem análises abrangentes de modelagem de ameaças para avaliar a prontidão para cargas previstas de jogadores e tentativas de acesso não autorizado, além de preparar procedimentos de resposta a incidentes.

Etapas de implementação

- Implemente grades de proteção em todas as etapas do seu CI/CD pipeline.
- Use automação e ferramentas para melhorar a eficiência das análises de segurança de seus aplicativos.
- Use a modelagem de ameaças como um processo para melhorar a segurança de seus aplicativos.

Recursos

- AWS Blog de segurança: Como abordar a modelagem de ameaças
- NIST: Guia para modelagem de ameaças de sistemas centrada em dados
- Modelagem de ameaças da maneira certa para criadores — treinamento individualizado virtual do AWS Skill Builder
- Modelagem de ameaças para construtores — Workshop AWS

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas relacionadas à segurança.

Documentos relacionados:

- Cenários comuns do Amazon Cognito
- Usando assinado URLs
- Use fluxos de canais para remover palavrões e conteúdo confidencial das mensagens no Amazon Chime SDK
- Segurança na Amazon GameLift
- Entrega segura de conteúdo usando a Amazon CloudFront
- Guia de respostas de segurança
- AWS Melhores práticas para resiliência DDo de S
- Protegendo seu Nuvem AWS ambiente contra ransomware

Soluções de parceiros relacionados:

- Datadog
- Sumo Logic
- Splunk
- Honeycomb.io
- New Relic
- AWS Marketplace - Soluções DRM

Materiais de treinamento relacionados:

- Conceitos básicos do Amazon Cognito
- Treinamento individualizado de segurança

Confiabilidade

O pilar de confiabilidade inclui a capacidade de um sistema se recuperar de interrupções na infraestrutura ou no serviço, adquirir dinamicamente recursos de computação para atender à demanda e mitigar interrupções, como configurações incorretas ou problemas transitórios de rede.

Áreas de foco

- Princípios de design
- Fundamentos
- Arquitetura da workload
- Gerenciamento de alterações
- Gerenciamento de falhas
- Recursos

Princípios de design

Além dos princípios de design no whitepaper do AWS Well-Architected Framework, a seguir estão os princípios de design que podem aumentar a confiabilidade na nuvem para cargas de trabalho de jogos:

- Estabeleça a linha de base para o pico de simultaneidade de jogadores e as metas de escalabilidade do sistema necessárias para atender às projeções de negócios: antes de lançar um jogo e durante as operações do jogo ao vivo, desenvolva estimativas para o número de jogadores simultâneos esperados no pico para estabelecer metas de escalabilidade do sistema para atender a essas projeções. Isso ajuda a criar uma linha de base para a confiabilidade do seu jogo. Defina políticas de escalabilidade para acomodar automaticamente as mudanças na demanda, sem afetar a disponibilidade, verificando se seus sistemas de escalabilidade gerenciam com facilidade as sessões ativas dos jogadores.
- Avalie sua confiabilidade e o impacto na experiência do jogador: defina os principais indicadores de desempenho (KPIs) que representem a saúde do seu jogo. Monitore o impacto das mudanças na infraestrutura e nos recursos do jogo em sua confiabilidade.

Fundamentos

Para alcançar a confiabilidade, um sistema deve ter uma base bem planejada e um monitoramento com mecanismos para lidar com mudanças na demanda ou nos requisitos. O sistema deve ser projetado para detectar falhas e se curar automaticamente.

Não há práticas recomendadas básicas específicas para o Games Lens. Consulte o whitepaper do Well-Architected Framework para obter as melhores práticas sobre fundamentos de confiabilidade que se aplicam às cargas de trabalho de jogos.

Arquitetura da workload

GAMEREL01: Sua arquitetura de jogo está aproveitando a resiliência da nuvem?

AWS a infraestrutura é construída em torno de regiões e zonas de disponibilidade. Regiões da AWS fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, que são conectadas usando redes de baixa latência, alta taxa de transferência e altamente redundantes. Essas construções podem ser usadas para arquitetar cargas de trabalho com metas de confiabilidade em foco.

Práticas recomendadas

- GAMEREL01-BP01 Distribua a infraestrutura de jogos em várias zonas e regiões de disponibilidade para melhorar a resiliência

GAMEREL01-BP01 Distribua a infraestrutura de jogos em várias zonas e regiões de disponibilidade para melhorar a resiliência

Para minimizar o impacto de deficiências localizadas na infraestrutura em seus jogadores, você deve distribuir sua implantação de infraestrutura uniformemente em locais independentes suficientes para ser capaz de resistir a deficiências inesperadas e, ao mesmo tempo, ter capacidade suficiente para atender às necessidades da demanda de seus jogadores.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao implantar sua infraestrutura de jogo, é recomendável distribuir uniformemente sua capacidade em várias zonas de disponibilidade em uma região para que você possa resistir a interrupções em uma ou mais zonas de disponibilidade sem interromper a experiência do jogador. Os serviços de back-end de jogos, como aplicativos web, devem ter balanceamento de carga em várias zonas de disponibilidade ou devem ser criados usando serviços gerenciados, como AWS Lambda o Amazon API Gateway, que fornece alta disponibilidade regional por design. Da mesma forma, componentes que mantêm o estado, como caches, bancos de dados, filas de mensagens e soluções de armazenamento, devem ser projetados para fornecer persistência durável de dados em várias zonas de disponibilidade, que é fornecida por design em serviços como Amazon S3, DynamoDB e Amazon SQS, e pode ser configurada em outros serviços.

Ao projetar sua arquitetura de hospedagem de servidores de jogos para oferecer resiliência, implante suas frotas de servidores de jogos uniformemente nas zonas de disponibilidade de um Região da AWS para maximizar seu acesso à capacidade computacional disponível na região, bem como reduzir o escopo do impacto das deficiências da zona de disponibilidade. Por exemplo, você pode configurar o Amazon EC2 Auto Scaling para usar as zonas de disponibilidade. Se uma EC2 instância não estiver íntegra, o EC2 Auto Scaling poderá substituí-la e executar instâncias em outras zonas de disponibilidade se uma ou mais zonas de disponibilidade ficarem indisponíveis.

Para infraestrutura crítica, como autenticação, provisione um número mínimo de instâncias viáveis em execução em várias zonas de disponibilidade e use o escalonamento automático para lidar com aumentos de carga ou tolerância a falhas se houver uma deficiência em uma das zonas de disponibilidade.

Implante sua infraestrutura de jogos em várias regiões para maximizar a disponibilidade. Recursos de recuperação de desastres entre regiões, como bancos de dados globais Aurora e infraestrutura redundante, que podem se tornar ativos com uma simples alteração de DNS implantada em uma região secundária, podem fornecer continuidade do serviço caso a região primária seja prejudicada. Embora incentivemos isso para que seus serviços de back-end de jogos alcancem alta disponibilidade, essa recomendação é especialmente importante para seus servidores de jogos.

Por exemplo, em um jogo multijogador, sua capacidade de infraestrutura para servidores de jogos provavelmente superará as necessidades de capacidade de seus outros serviços, já que os servidores de jogos são usados para hospedar sessões de jogos para jogadores. Muitos jogos optam por dividir os jogadores em regiões lógicas (como oeste e leste dos EUA). Para simplificar a experiência do jogador e facilitar o uso da infraestrutura global para hospedar jogos, considere

desacoplar o nome de suas regiões de jogo voltadas para jogadores da região do provedor de nuvem subjacente ou da localização do data center que hospeda fisicamente os servidores do jogo, junto com outras infraestruturas, como Zonas Locais ou seus próprios data centers que hospedam instâncias de servidores de jogos que oferecem suporte a essa região de jogo do jogador.

Ao projetar seu serviço de matchmaking, implante uma arquitetura multirregional com implantações de software separadas em todas as regiões. Separe sua implantação de serviço de matchmaking das frotas que hospedam suas instâncias de servidor de jogo para que você possa direcionar jogadores para um servidor de jogo em regiões, independentemente de qual implantação regional de seu serviço de matchmaking atendeu à solicitação de matchmaking.

Crie uma lógica em sua implementação de matchmaking para favorecer as regiões do servidor de jogo que atendam à sua latência e outras regras, com a capacidade de voltar a rotear jogadores para outras regiões se suas frotas estiverem com pouca capacidade ou se houver outras interrupções na infraestrutura regional.

Etapas de implementação

- Distribua a infraestrutura de jogos uniformemente em várias zonas de disponibilidade para fornecer alta disponibilidade e resiliência.
- Implante serviços de back-end de jogos e componentes com monitoramento de estado usando soluções gerenciadas como Amazon S3 AWS Lambda, DynamoDB e SQS, ou configure o balanceamento de carga e a durabilidade para soluções personalizadas.
- Implemente implantações em várias regiões para serviços e servidores essenciais de jogos, usando soluções de recuperação de desastres, como bancos de dados globais Aurora e regiões lógicas voltadas para jogadores, separadas das localizações físicas subjacentes.

Recursos

- Estabilidade estática
- Melhores práticas para filas de sessões de GameLift jogos da Amazon
- Frotas GameLift multirregionais da Amazon
- Banco de dados global Aurora para recuperação de desastres

Gerenciamento de alterações

GAMEREL02: Como você dimensiona seus jogos com estado para acomodar mudanças na demanda?

À medida que a demanda de seus jogadores flutua com o tempo, sua infraestrutura de jogo deve ser capaz de se expandir de forma adaptativa para lidar com essas mudanças nos requisitos. Embora seja difícil prever a popularidade de um jogo com antecedência, crie uma abordagem de arquitetura que permita a adição e remoção da capacidade da infraestrutura para acomodar as flutuações na população de jogadores.

Práticas recomendadas

- GAMEREL02-BP01 Implemente uma estratégia de escalonamento que incorpore o estado das sessões de jogo ativas dos jogadores
- GAMEREL02-BP02 Support o uso de vários tipos de instância para seu jogo EC2

GAMEREL02-BP01 Implemente uma estratégia de escalonamento que incorpore o estado das sessões de jogo ativas dos jogadores

Implemente uma solução para escalar automaticamente sua infraestrutura de jogo de uma forma que incorpore a natureza estável de suas sessões de jogadores ativamente conectados e gerencie as atividades de escalabilidade com elegância sem interromper a jogabilidade.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Uma das vantagens de desenvolver um jogo na nuvem é a elasticidade que pode ser obtida ao escalar automaticamente a infraestrutura do servidor conforme necessário para atender à demanda. Embora jogos e serviços de back-end sem estado ou assíncronos possam ser escalados dinamicamente usando as políticas do Amazon Auto EC2 Scaling, o escalonamento automático EKS ou técnicas similares normalmente adotadas para aplicativos web escaláveis, os desenvolvedores de jogos geralmente precisam de uma abordagem mais personalizada para escalar jogos síncronos ou com estado para ajudar a bloquear interrupções nas sessões ativas dos jogadores.

Etapas de implementação

- Para jogos com estado, gere métricas personalizadas que possam ser usadas para monitorar o estado das sessões de seus jogadores e a capacidade disponível do servidor de jogos, que podem ser reportadas à Amazon CloudWatch como métricas personalizadas. Exercite recursos com monitoramento de aplicativos, como o CloudWatch Synthetics, que verificam se o jogo está comprometido em funções que um simples monitoramento de integridade de alta e baixa pode não detectar.
- Usando métricas personalizadas, implemente um software de escalabilidade de servidor de jogos, por exemplo, como um aplicativo sem servidor usando a AWS Lambda função ou AWS Fargate, para gerenciar a frota de instâncias dedicadas do servidor de jogos, usando o AWS SDK para fazer chamadas de API para atualizar as configurações de capacidade mínima, máxima e desejada para os grupos de Auto Scaling EC2 que hospedam sua criação de servidor de jogos.
- Use GameLift a Amazon para hospedar seus servidores de jogos e use os recursos de escalabilidade automática do servidor de out-of-the-box jogos para gerenciar esse processo de escalabilidade para você.

Os recursos de escalabilidade automática da Amazon GameLift estão cientes das sessões ativas dos jogadores e podem ser configurados para bloquear o encerramento ou a expansão de instâncias do servidor de jogos que hospedam jogadores ativamente. Para obter mais informações, consulte Monitorar GameLift servidores da Amazon com a Amazon CloudWatch.

GAMEREL02-BP02 Support o uso de vários tipos de instância para seu jogo EC2

Ao hospedar seu jogo usando EC2 instâncias, ou se você usar contêineres hospedados em EC2 instâncias no seu Conta da AWS, use vários tipos de instância em sua estratégia de hospedagem. Ao usar vários tipos de instância, você pode aumentar o número de opções de computação que podem ser usadas quando seu jogo está sendo escalado para adicionar mais servidores para apoiar o crescimento do número de jogadores, o que melhora a confiabilidade caso seu tipo de instância preferido não esteja disponível.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao hospedar seu jogo usando Instâncias Spot, use vários tipos de instância, pois a disponibilidade das Instâncias Spot varia de acordo com a demanda do cliente.

Teste seu jogo em vários tipos de instância para atender aos requisitos de custo e desempenho e determinar uma classificação priorizada dos tipos de instância. O Amazon EC2 Auto Scaling suporta o uso de vários tipos e tamanhos de instância, bem como a atribuição de pesos a cada tipo de instância em sua configuração para que você possa implementar uma classificação priorizada de opções de computação.

Ao hospedar seu jogo usando a hospedagem GameLift gerenciada da Amazon, decida que tipo de instâncias seu jogo precisa e como executar os processos do servidor de jogos nelas (usando uma configuração de tempo de execução). Ao escolher recursos para uma frota, considere vários fatores, incluindo sistema operacional do jogo, tipo de instância (o hardware de computação) e se deve usar instâncias sob demanda, instâncias spot ou ambas. Os custos de hospedagem com a Amazon dependem GameLift principalmente do tipo de instância que você usa. Para obter mais informações, consulte Escolher recursos de computação para uma frota gerenciada.

Etapas de implementação

- Use vários tipos de EC2 instância para melhorar a confiabilidade e as opções de escalabilidade ao hospedar seu jogo em EC2 nossos contêineres.
- Configure o Amazon EC2 Auto Scaling ou GameLift frotas com tipos e pesos de instância priorizados para otimizar o custo e o desempenho.
- Teste seu jogo em vários tipos de instância para verificar se o desempenho atende aos requisitos e ajuste sua estratégia de hospedagem adequadamente .

Gerenciamento de falhas

GAMEREL03: Como você mantém o estado do jogo durante interrupções na infraestrutura?

Como a infraestrutura do jogo passa por vários eventos operacionais ao longo do tempo, a arquitetura do jogo deve ser projetada para manter a continuidade das experiências dos jogadores e preservar o estado do jogo durante os eventos de infraestrutura. Para lidar com esses eventos, implemente mecanismos de monitoramento, desligamentos regulares e persistência de estado para garantir experiências de jogo tranquilas para seus jogadores.

Práticas recomendadas

- GAMEREL03-BP01 Monitore interrupções no servidor de jogos e use os dados para melhorar a arquitetura de hospedagem e atingir metas de confiabilidade
- GAMEREL03-BP02 Implemente um acoplamento flexível de recursos do jogo para lidar com falhas com impacto mínimo na experiência do jogador
- GAMEREL03-BP03 Monitore eventos de infraestrutura ao longo do tempo para medir o impacto no comportamento do jogador

GAMEREL03-BP01 Monitore interrupções no servidor de jogos e use os dados para melhorar a arquitetura de hospedagem e atingir metas de confiabilidade

Monitore as métricas do servidor de jogos e o impacto de falhas ou degradações no desempenho, como aumento da latência sob carga, no comportamento do jogador ao longo do tempo, para que você possa ajustar sua estratégia de hospedagem de servidores de jogos para atender aos requisitos de confiabilidade do jogo. A infraestrutura do servidor de jogos a ser degradada deve ser removida do serviço imediatamente se estiver afetando os jogadores ou substituída proativamente quando não houver sessões ativas de jogadores hospedadas no servidor.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Para cenários em que os jogos são hospedados como REST APIs, a confiabilidade do sistema pode ser gerenciada como as arquiteturas tradicionais de aplicativos da Web, em que o tráfego pode ser balanceado em vários servidores de forma distribuída para reduzir o risco de falhas no servidor.

Para jogabilidade síncrona em tempo real, uma sessão de jogo geralmente é hospedada em um processo de servidor de jogos executado em uma máquina virtual ou instância de servidor de jogos, pois o estado do jogo precisa ser mantido de forma eficiente e replicado para clientes de jogos conectados. Essa implementação significa que a experiência do jogador está fortemente associada ao desempenho e à confiabilidade do processo do servidor de jogos que hospeda sua sessão de jogo. Esse tipo de arquitetura torna o gerenciamento da confiabilidade dos servidores de jogos mais complexo do que as abordagens tradicionais.

Para mitigar o impacto de uma falha no servidor do jogo, configure seu jogo para realizar continuamente atualizações assíncronas do estado do jogo de um jogador em um cache ou banco

de dados de alta disponibilidade, como Amazon (ElastiCache Redis OSS) ou Amazon MemoryDB. Se ocorrer uma falha no servidor, o último estado de jogo salvo do jogador poderá ser obtido no armazenamento de dados externo e sua sessão poderá ser restaurada em uma nova instância do servidor de jogos.

No entanto, essa abordagem adiciona custo e complexidade adicionais ao gerenciamento desse estado externo e pode não ser adequada para jogos competitivos ou de ritmo acelerado em que as mudanças de estado são tão frequentes e ocorrem em uma escala tão significativa que a introdução até mesmo de um armazenamento de dados em cache na memória de alto desempenho resultaria em um atraso de replicação significativo demais para ser útil restaurar uma sessão. Para jogos dessa natureza, a abordagem ideal é aceitar a perda do servidor e enviar o jogador de volta ao lobby do jogo para encontrar outra sessão ou você pode redirecioná-lo automaticamente para outra sessão de jogo.

Capture o máximo de dados de registro úteis sobre o que causou a interrupção do servidor para que você possa investigar o problema posteriormente. GameLift A Amazon fornece orientação para depuração de problemas da frota e fornece a capacidade de acessar remotamente as instâncias da frota da Amazon GameLift .

Etapas de implementação

- Monitore as métricas do servidor de jogos em busca de degradações de desempenho e remova ou substitua servidores degradados conforme necessário para manter a confiabilidade.
- Use a Amazon ElastiCache ou o MemoryDB para atualizações assíncronas do estado do jogo para permitir a recuperação da sessão após falhas no servidor, quando possível.
- Capture dados de registro detalhados sobre interrupções no servidor para investigação e depuração, utilizando ferramentas como a GameLift Amazon para monitoramento de frotas e acesso remoto.

GAMEREL03-BP02 Implemente um acoplamento flexível de recursos do jogo para lidar com falhas com impacto mínimo na experiência do jogador

Os componentes de desacoplamento se referem ao conceito de projetar componentes do servidor para que possam operar da forma mais independente possível. Alguns aspectos do jogo são difíceis de separar, pois os dados precisam estar o mais atualizados possível para proporcionar uma boa experiência de jogo aos jogadores. No entanto, muitos componentes e tarefas de jogo podem ser dissociados. Por exemplo, tabelas de classificação e serviços de estatísticas não são essenciais

para a experiência de jogo, e as leituras e gravações nesses serviços podem ser realizadas de forma assíncrona a partir do jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Implemente uma degradação suave de recursos em seu jogo que podem ser desativados automaticamente ou por um administrador se forem detectados problemas, bem como configure serviços upstream que dependem do recurso para serem capazes de lidar com a falha sem problemas. Por exemplo, se dados específicos do jogador não estiverem sendo carregados corretamente em seu cliente de jogo, considere se esses dados são essenciais para a experiência de jogo. Caso contrário, configure o cliente do jogo para lidar com essa falha sem interromper a experiência do jogador, optando por tentar buscar novamente esses dados mais tarde, quando o jogador revisitar a tela.

Use lógica como tempos limite, novas tentativas e recuos para lidar com erros e falhas. Os tempos limite evitam que os sistemas travem por períodos excessivamente longos. As novas tentativas podem fornecer alta disponibilidade de erros transitórios e aleatórios.

Defina componentes não críticos que podem ser fracamente acoplados a componentes críticos. O acoplamento frouxo permite que os sistemas sejam mais resilientes, pois a falha em um componente não ocorre em cascata em outros. Quando os recursos do jogo não exigem conexões com estado com seus servidores ou back-end de jogos, você deve implementar protocolos sem estado para escalar dinamicamente e se recuperar de falhas transitórias. Desenvolva seus componentes não críticos onde eles possam ser facilmente acoplados a protocolos sem estado usando uma API. HTTP/JSON Implemente chamadas de rede do cliente do jogo de forma assíncrona e sem bloqueio para minimizar o impacto nos jogadores de recursos de jogo de baixo desempenho ou outros serviços dependentes.

Para melhorar ainda mais a resiliência por meio de acoplamento frouxo, use um serviço de mensagens, como filas, streaming ou um sistema baseado em tópicos entre componentes que possam ser manipulados de forma assíncrona. Esse modelo é adequado para uma interação que não exige uma resposta imediata ou em que a confirmação de que uma solicitação foi registrada é suficiente. Essa solução envolve um componente que gera eventos e outro que os consome. Os dois componentes não se integrarão por meio de point-to-point interação direta, mas por meio de um intermediário, como uma camada durável de armazenamento ou enfileiramento. Isso também ajuda a melhorar a confiabilidade do sistema, preservando as mensagens quando o processamento falha.

Pesquise e selecione um mecanismo de mensagens apropriado, pois vários serviços de mensagens têm características diferentes, como mecanismos de pedido e entrega. Projete operações de forma que sejam idempotentes para que o sistema de mensagens escolhido entregue mensagens pelo menos uma vez. Como exemplo, considere um caso de uso típico de um jogo em que seu jogo precisa monitorar o tempo de jogo, as estatísticas ou outros dados relevantes do jogador, o que pode levar a um caso de uso de alta taxa de gravação em momentos de pico de simultaneidade de jogadores.

Para implementar uma arquitetura confiável, considere se o caso de uso exige read-after-write consistência conforme percebida pelo jogador. Normalmente, cenários como esses são adequados para processamento assíncrono e podem ser alcançados implementando um padrão de filas de gravação em que as solicitações são ingeridas em uma fila de mensagens escalável e durável, como o Amazon SQS, e podem ser inseridas em seu banco de dados de back-end em lotes usando um serviço ao consumidor, como uma função Lambda. Essa abordagem é mais confiável do que a comunicação síncrona entre vários componentes distribuídos, incluindo o cliente de jogo do jogador, seus servidores de aplicativos e web de back-end e seu sistema de banco de dados interno. Também reduz os custos porque o banco de dados de back-end não precisa ser escalado para atingir o pico de taxa de transferência de gravação, pois o processamento do consumidor na fila de gravação pode ser usado para diminuir essa taxa de ingestão conforme necessário.

Etapas de implementação

- Separe componentes não críticos, como tabelas de classificação e serviços de estatísticas, de recursos essenciais de jogabilidade para permitir operações assíncronas e aumentar a resiliência.
- Implemente uma degradação suave para recursos não críticos com lógica para tempos limite, novas tentativas e recuos, e verifique se o cliente do jogo lida com falhas sem interromper a experiência do jogador.
- Use sistemas de mensagens como o Amazon SQS para comunicação assíncrona entre componentes, permitindo o processamento escalável, durável e confiável de casos de uso de alto rendimento.

Recursos

- Crie cargas de trabalho altamente escaláveis e confiáveis usando a arquitetura de microsserviços
- Integrando microsserviços usando serviços sem servidor AWS
- Entendendo o sistema de mensagens assíncronas para microsserviços
- Introdução aos padrões escaláveis de desenvolvimento de jogos em AWS

- Implementando uma degradação elegante

GAMEREL03-BP03 Monitore eventos de infraestrutura ao longo do tempo para medir o impacto no comportamento do jogador

Monitore o processo do servidor de jogos, as métricas da instância do servidor de jogos e as métricas de experiência do jogo para determinar a causa raiz dos problemas. Além de monitorar a CPU e a memória, você também pode configurar o monitoramento de métricas de rede relacionadas às limitações de rede das EC2 instâncias para alertá-lo sobre problemas como excesso de largura de banda ou outros problemas no nível da rede que possam indicar que os recursos do seu servidor estão subprovisionados. packets-per-second

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Use o CloudWatch Synthetics para verificar a funcionalidade do aplicativo de caminhos críticos para a experiência do jogador, como a incapacidade de fazer login ou outros problemas que afetem o serviço. Para servidores de jogos hospedados usando a Amazon GameLift, considere monitorar métricas como:

- GameServerInterruptions e InstanceInterruptions, o que pode ajudar a entender como as limitações na disponibilidade de instâncias Spot estão afetando seus servidores de jogos implantados usando o Spot.
- ServerProcessAbnormalTerminations, que pode ser usado para detectar terminações anormais nos processos do servidor do jogo.

É recomendável manter os dados históricos das métricas da confiabilidade do seu servidor de jogos. Use esses dados históricos para fins de geração de relatórios e junte-os a outros conjuntos de dados para descobrir possíveis tendências e avaliar o impacto no comportamento do jogador ao longo do tempo que pode ser devido a problemas no servidor do jogo.

CloudWatch A Amazon não retém métricas indefinidamente, e a resolução de armazenamento das métricas aumenta com o tempo, então considere exportar essas métricas para um armazenamento econômico de longo prazo, como o Amazon S3. Você pode configurar o CloudWatchMetric Streams para entregar automaticamente suas métricas CloudWatchdas regiões para seu próprio bucket do S3, onde elas podem ser armazenadas a longo prazo em um nível de armazenamento, como o S3

Intelligent-Tiering, e, eventualmente, arquivadas usando o Amazon Glacier. Ao colocar suas métricas no Amazon S3, elas estão prontamente disponíveis para serem unidas a outros conjuntos de dados em seu data lake para consultas interativas com o Amazon Athena.

Etapas de implementação

- Monitore métricas de servidores, instâncias e redes de jogos, incluindo largura de banda e packet-per-second limites, usando a Amazon CloudWatch e a CloudWatch Synthetics para verificar a funcionalidade de caminhos críticos.
- Monitore métricas GameLift específicas, como GameServerInterruptionse ServerProcessAbnormalTerminationspara avaliar o impacto da disponibilidade de instâncias spot e detectar terminações anormais do servidor.
- Exporte CloudWatch métricas para o Amazon S3 para armazenamento de longo prazo, use níveis econômicos, como S3 Intelligent-Tiering ou Glacier, e analise tendências com ferramentas como o Amazon Athena.

Recursos

- As métricas de desempenho de rede EC2 em nível de instância da Amazon revelam novos insights
- CloudWatchMetric Streams — Envie AWS métricas para parceiros e para seus aplicativos em tempo real

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas relacionadas à confiabilidade.

Documentos relacionados:

- Praticando a integração contínua e a entrega contínua em AWS
- Escalonamento automático de filas de trabalhos assíncronas
- Projete sua WorkloadService arquitetura
- Tempos limite, novas tentativas e recuo com instabilidade
- Well-Architected Framework - Pilar de confiabilidade
- Arquitetura para escalabilidade confiável
- A biblioteca do Amazon Builder

- Mensagens em tempo real em grande escala para jogos multijogador
- Introdução aos padrões escaláveis de desenvolvimento de jogos em AWS
- Executando microsserviços em contêineres em AWS
- Hospedagem de aplicativos Web na nuvem
- Construindo uma infraestrutura de rede multi-VPC escalável e segura

Vídeos relacionados:

- re:Invent 2020: Ubisoft: Construindo um jogo multijogador multiplataforma em AWS
- re:Invent 2018: Supercell — Jogos móveis em grande escala
- re:Invent 2019: Como a CAPCOM cria jogos divertidos com contêineres, dados e ML
- re:Invent 2018: Globalizando as contas de jogadores na Riot Games e mantendo a disponibilidade
- re:Invent 2020: GameLoft - Um mergulho profundo sobre a migração do data lake com tempo de inatividade zero

Treinamento relacionado:

- Usando o Amazon GameLiftFleet IQ para servidores de jogos
- Hospedagem de servidores de jogos com a Amazon EC2

Eficiência de performance

O pilar de eficiência de desempenho se concentra no uso eficiente dos recursos de computação para atender aos requisitos e manter essa eficiência à medida que a demanda muda e as tecnologias evoluem.

Adote uma abordagem baseada em dados para selecionar uma arquitetura de alto desempenho. Reúna dados abrangentes sobre a arquitetura, desde o design de alto nível até a seleção e configuração dos tipos de recursos. Considere suas escolhas arquitetônicas de forma cíclica para aproveitar o conjunto de serviços e soluções em constante evolução. As métricas ajudam a entender o desvio do desempenho esperado para que você possa agir. Uma abordagem baseada em dados ajuda a fazer concessões com sua arquitetura para melhorar o desempenho, reduzir custos ou melhorar a experiência do desenvolvedor.

Áreas de foco

- Princípios de design
- Seleção de arquitetura
- Seleção da região
- Desenvolvimento iterativo
- Computação e hardware
- Seleção de computação
- Gerenciamento de dados
- Rede e entrega de conteúdo
- Processo e cultura
- Recursos

Princípios de design

Além dos princípios de design no whitepaper do AWS Well-Architected Framework, os seguintes princípios de design podem alcançar eficiência de desempenho para seus jogos:

- Meça o desempenho do jogo a partir de end-to-end: É importante medir o desempenho conforme ele é percebido da perspectiva de seus jogadores. Isso significa que você deve medir o desempenho do cliente do jogo, sua infraestrutura de jogo e a conectividade com a Internet que

conecta seus jogadores à infraestrutura. Isso ajudará a entender onde você pode fazer melhorias de desempenho em sua arquitetura.

- Otimize sua arquitetura para melhorar as métricas que refletem a experiência real do jogador: à medida que você adapta e evolui sua arquitetura ao longo do tempo, pense em como essas melhorias e mudanças afetarão a experiência do jogador. As cargas de trabalho de jogos devem ser capazes de suportar e minimizar o impacto das falhas para bloquear interrupções generalizadas na jogabilidade. Os recursos e sistemas do jogo que não dependem criticamente uns dos outros devem ser desacoplados para reduzir o raio de explosão das falhas e isolar os problemas que afetam os jogadores.
- Use tecnologias que simplifiquem as operações do jogo e aumentem a velocidade de desenvolvimento: priorize a adoção de tecnologias que possam melhorar a eficiência do desenvolvedor. A sobrecarga operacional durante os estágios de desenvolvimento de pré-produção pode ser uma distração da melhoria da jogabilidade. Ao aproveitar os serviços gerenciados de AWS nossos AWS parceiros, a engenharia pode ser reduzida, o que permite que os desenvolvedores de jogos se concentrem no ciclo principal do jogo e na experiência do jogador. Os requisitos de arquitetura e desempenho podem mudar e evoluir ao longo do ciclo de vida do desenvolvimento do jogo, e as compensações tecnológicas devem ser consideradas em cada fase.
- Projete a infraestrutura para atender ao pico de simultaneidade de jogadores e escale dinamicamente conforme necessário: a infraestrutura deve ser projetada para ser dimensionada para atender à demanda dos jogadores. Métricas, como simultaneidade de sessões de jogadores e número de logins, podem ser usadas para escalar preventivamente antes que os sistemas fiquem sobrecarregados. Métricas reativas de utilização do sistema, como consumo de CPU e memória, podem ser usadas para escalar depois que os sistemas ficarem sobrecarregados. Ao escalar dinamicamente sua infraestrutura, você pode reduzir os custos de operação do seu jogo.

Seleção de arquitetura

GAMEPERF01: Como você seleciona a opção de hospedagem apropriada para seus servidores de jogos?

A seleção da opção de hospedagem apropriada para seus servidores de jogos é fundamental para o desempenho do servidor de jogos. A decisão de usar EC2 instâncias, uma solução de contêiner ou um serviço totalmente gerenciado é uma das primeiras decisões a serem tomadas ao arquitetar

para produção. Cada opção de hospedagem terá recursos e considerações variados para ajuste de desempenho, escalabilidade, operações e integração.

Práticas recomendadas

- GAMEPERF01-BP01 Avalie os requisitos de recursos e as necessidades de escalabilidade do servidor de jogos
- GAMEPERF01-BP02 Considere a sobrecarga operacional para escalar servidores de jogos
- GAMEPERF01-BP03 Avalie a integração com outros AWS serviços, ambientes de desenvolvimento, arquiteturas de CPU de destino e recursos

GAMEPERF01-BP01 Avalie os requisitos de recursos e as necessidades de escalabilidade do servidor de jogos

Avalie os requisitos do servidor em relação às suas necessidades de escalabilidade para verificar se você está selecionando uma opção de hospedagem que atenda aos seus requisitos e ofereça desempenho ideal.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao selecionar a opção de hospedagem apropriada para seus servidores de jogos, considere os seguintes fatores:

Requisitos de recursos do servidor de jogos

Avalie os requisitos de CPU, memória, rede e armazenamento dos processos do servidor de jogos para determinar o que o jogo consome. Não negligencie a rede; cada quadro requer ciclos de CPU para receber as ações do jogador, atualizar o estado do jogo e enviá-lo de volta ao jogador. O descarregamento do processamento de pacotes pode liberar a CPU para as funções principais do jogo. A rede é a base para uma jogabilidade suave e responsiva, portanto, testá-la logo no início do processo define um perfil básico de desempenho para um jogo.

Um jogo de tiro em primeira pessoa pode ter altas ações por segundo, que a CPU precisa transferir rapidamente para a rede, o que pode favorecer instâncias da família C otimizadas para computação, enquanto um jogo de estratégia baseado em turnos, que pode passar mais ciclos de CPU processando por turno, pode precisar de mais memória das instâncias da família R para armazenar

e atualizar localmente o estado do jogo no servidor antes de enviá-lo de volta aos jogadores. Use uma abordagem baseada em dados, como o Método de Saturação e Erros de Utilização (USE), para fazer escolhas arquitetônicas bem informadas.

Escalabilidade e elasticidade

Avalie com que rapidez e facilidade cada opção de hospedagem pode ser escalada para atender à demanda dos jogadores sem comprometer o desempenho. Considere o nível de automação e flexibilidade necessário para que a carga de trabalho do seu jogo mantenha uma experiência de jogo tranquila durante os horários de pico. Um servidor de jogos pode ser escalado rapidamente aumentando a utilização por meio da adição de processos adicionais de servidor de jogos na mesma instância, em que um back-end de jogos pode ser escalado mais lentamente com base no aumento do número de usuários ativos e nos jogos que estão sendo jogados. Sua frota deve ser dimensionada de acordo com a demanda para minimizar os custos e, ao mesmo tempo, facilitar o tempo mínimo de espera para os jogadores entrarem no jogo. Analise o Amazon EC2 Spot Instance Advisor para obter informações sobre a capacidade econômica disponível para frotas de servidores de jogos.

Etapas de implementação

- Avalie os requisitos de recursos do servidor de jogos para CPU, memória, rede e armazenamento para selecionar os tipos de instância adequados, considerando as necessidades de desempenho específicas do jogo, como alta taxa de transferência de rede para jogos de FPS ou otimização de memória para jogos de estratégia baseados em turnos.
- Compare diferentes opções de hospedagem, como contêineres, instâncias, bare-metal e serviços gerenciados, analisando dados de desempenho usando estruturas como o método USE. Use esses insights para tomar melhores decisões sobre a arquitetura do seu sistema.
- Projete frotas para escalabilidade e elasticidade, utilizando ferramentas como o EC2 Spot Instance Advisor para otimizar custos e, ao mesmo tempo, facilitar o escalonamento rápido para atender à demanda dos jogadores nos horários de pico.

GAMEPERF01-BP02 Considere a sobrecarga operacional para escalar servidores de jogos

Considere a sobrecarga operacional e de gerenciamento associada a cada opção de hospedagem.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Sobrecarga operacional

Soluções auto-hospedadas em EC2 ou contêineres podem fornecer mais controle, mas também exigirão mais gerenciamento. Orquestradores de contêineres como ECS ou EKS podem reduzir os tempos de lançamento de servidores em contêineres e, ao mesmo tempo, aumentar a complexidade da rede e a sobrecarga de orquestração de manutenção.

Por exemplo, os grupos de nós gerenciados pelo EKS podem automatizar o provisionamento e o gerenciamento do ciclo de vida de seus servidores de jogos, mas não respeitam os orçamentos de interrupção do pod ao encerrar um nó. Se seu jogo exigir mais do que o período de encerramento de 15 minutos para concluir os jogos com segurança, talvez seja necessário criar ganchos de ciclo de vida ou considerar nós autogerenciados com controladores personalizados para bloquear a interrupção do jogo.

Serviços gerenciados como o Amazon Game Lift podem lidar com a maior parte da sobrecarga operacional, mas reduzir a quantidade de visibilidade e controle sobre os requisitos especiais de configuração de rede e segurança de baixo nível. Escolher uma solução de servidor de jogos é uma troca entre o nível de personalização, controle e responsabilidade que você terá por ajustar o desempenho do servidor de jogos e o comportamento de escalabilidade.

Etapas de implementação

- Avalie a sobrecarga operacional das opções de hospedagem, equilibrando o esforço de controle e gerenciamento entre soluções auto-hospedadas EC2, como ECS ou EKS, e serviços gerenciados, como o Amazon Game Lift.
- Use grupos de nós gerenciados pelo EKS para automação, mas implemente ganchos de ciclo de vida ou controladores personalizados se seus servidores de jogos exigirem períodos de encerramento mais longos do que o padrão.
- Avalie as compensações entre personalização, visibilidade e responsabilidade operacional ao selecionar uma solução de servidor de jogos.

GAMEPERF01-BP03 Avalie a integração com outros AWS serviços, ambientes de desenvolvimento, arquiteturas de CPU de destino e recursos

Avalie o quão bem cada opção de hospedagem se integra a outros AWS serviços dos quais seu jogo depende, como bancos de dados, análises ou serviços de entrega de conteúdo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Integração com outros serviços da AWS

A integração perfeita entre os serviços oferece benefícios operacionais, como monitoramento aprimorado do desempenho e entrega eficiente e segura de dados entre componentes de jogos, servidores de jogos, serviços de back-end de jogos e soluções de observabilidade.

Por exemplo, coordenar turnos de trânsito para jogos ao vivo pode ser complexo. O Amazon Route 53 ajudará a manter seus registros de DNS atualizados, o que simplifica as mudanças de tráfego coordenadas. AWS As discagens de tráfego do Global Accelerator permitem que você envie uma porcentagem do tráfego para outra região e mantenha seu jogo funcionando durante a manutenção.

Ambiente e ferramentas de desenvolvimento

Considere as ferramentas de desenvolvimento, as estruturas e os ambientes compatíveis com cada opção de arquitetura. Verifique se a opção escolhida está alinhada à solução de desenvolvimento de jogos e às linguagens de programação, pois isso pode afetar a capacidade da sua equipe de otimizar e manter o desempenho do servidor de jogos. A entrega de um jogo em dispositivos móveis, consoles e PC aumentará a complexidade das ferramentas e dos testes. O suporte entre sistemas é particularmente importante para estúdios de vários jogos, onde serviços centralizados podem padronizar as melhores práticas de desenvolvimento em todos os títulos.

Arquitetura e recursos da CPU alvo

Considere o perfil de desempenho do seu motor de jogo e dos processos do servidor de jogos e o nível de suporte ARM disponível. Avalie se você pode se beneficiar da melhor relação preço/desempenho dos processadores baseados em Graviton ou x86 baseados em AMD64 ARM. Você precisa usar recursos da Intel, como criptografia AES-NI, AVX ou Turbo Boost? Analise os tipos de host dedicado para identificar famílias de instâncias de um ou vários soquetes. Ao usar uma família de instâncias com vários soquetes, considere a fixação NUMA e o compartilhamento de cache L3 nos processos do seu servidor de jogos. Use as configurações C-state e P-state para obter o melhor desempenho do seu jogo ajustando o relógio de frequência e reduzindo os níveis de sono.

Etapas de implementação

- Selecione opções de hospedagem com integração perfeita com AWS serviços como AWS Secrets Manager ACM e outros para ajudar a otimizar o monitoramento do desempenho, proteger a entrega de dados e reduzir as tarefas operacionais manuais.

- Verifique a compatibilidade entre sua opção de hospedagem e seu ambiente de desenvolvimento, estruturas e linguagens de programação para otimizar e manter o desempenho do servidor de forma eficaz.
- Avalie os requisitos de arquitetura de CPU, aproveitando o Graviton para relação preço-desempenho ou o x86 para recursos específicos, como AES-NI, AVX e Turbo Boost, e otimize o desempenho do servidor com fixação NUMA e ajuste C-state/P-state.

Seleção da região

GAMEPERF02: Como você determina quais regiões geográficas hospedar sua infraestrutura de jogos?

Escolher o local ideal para sua infraestrutura de jogos pode melhorar o desempenho da rede para jogadores e back-end. Considerar de onde sua base de jogadores está se conectando e como suas comunidades ou servidores são construídos é importante para o crescimento e a sustentabilidade a longo prazo em uma região geográfica. A implantação de infraestrutura de servidor de jogos e serviços de back-end desacoplados pode beneficiar sua eficiência operacional geral e melhorar a resiliência usando várias regiões, zonas locais e postos avançados para hospedar seu jogo.

Práticas recomendadas

- GAMEPERF02-BP01 Selecione uma região de origem próxima aos seus jogadores
- GAMEPERF02-BP02 Crie uma abordagem que ofereça suporte à colocação de uma infraestrutura de jogo sensível à latência perto dos jogadores para melhorar o desempenho

GAMEPERF02-BP01 Selecione uma região de origem próxima aos seus jogadores

Para o lançamento inicial de um jogo, você deve determinar onde implantar a infraestrutura com base nas discussões com as partes interessadas da empresa, como equipes de publicação que determinam onde se espera que o jogo seja disponibilizado aos jogadores e onde eles estão concentrando seus esforços de marketing e publicidade antes do lançamento.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

As partes interessadas de sua empresa também devem ter mecanismos para estimular a demanda para obter uma melhor compreensão da recepção e viabilidade dos jogadores. Por exemplo, essas equipes terão mecanismos como pré-venda de jogos, eventos e campanhas de marketing, listas públicas de e-mail para que os jogadores registrem interesse antes do lançamento e outras abordagens para estabelecer sinais relevantes para determinar onde o jogo provavelmente terá mais jogadores no lançamento. O jogo também pode usar uma estratégia de lançamento regional que inclui fases de teste de jogo e lançamento suave para determinar a demanda regional dos jogadores.

Selecione uma região de origem que esteja próxima à sua base de jogadores e de seus desenvolvedores e tenha os AWS serviços e recursos necessários para hospedar seu jogo. A casa RSegion será onde os serviços de back-end do jogo serão executados e também poderá executar servidores de jogos. Avalie uma região de origem com base nos serviços suportados, na conectividade com os pontos de presença, na proximidade das regiões de failover e no número de zonas de disponibilidade. Se você estiver usando uma zona local, considere que a região principal às vezes está localizada em uma área geográfica diferente. Por exemplo: a Zona Local us-east-1-scl-1a de Santiago, Chile, tem o Norte us-east-1 como sua região mãe, embora esteja geograficamente mais próxima de São Paulo sa-east-1.

Etapas de implementação

- Identifique regiões de implantação com base nos sinais de demanda dos jogadores provenientes de atividades de pré-lançamento, como pré-vendas, campanhas de marketing e registros de interesse.
- Escolha uma região de origem próxima à base principal de jogadores e desenvolvedores, certificando-se de que ela ofereça suporte AWS aos serviços necessários, pontos de presença e regiões de failover.
- Avalie cuidadosamente as Zonas Locais, considerando que a Região-mãe pode diferir geograficamente da localização da Zona Local.

GAMEPERF02-BP02 Crie uma abordagem que ofereça suporte à colocação de uma infraestrutura de jogo sensível à latência perto dos jogadores para melhorar o desempenho

O posicionamento separado para infraestrutura sensível à latência, como servidores de jogos, minimiza o impacto de longas rotas de rede. Implantações repetíveis podem simplificar a

manutenção de vários locais com melhor desempenho para seus jogadores. O ping é uma métrica comum que aparece na interface do jogo e o ping baixo pode ser uma capacidade diferenciadora.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao lançar um jogo pela primeira vez, talvez você ainda não tenha informações suficientes sobre sua base de jogadores para saber adequadamente onde implantar a infraestrutura mais próxima dos jogadores que estão mais interessados em jogar seu jogo. Esse é um desafio comum, e você deve se preparar para esse cenário projetando uma arquitetura que permita ajustar rapidamente sua estratégia de posicionamento de hospedagem para implantar servidores onde eles são necessários, mais perto dos jogadores. É comum que os desenvolvedores de jogos avaliem regularmente a implantação da infraestrutura de jogos como uma análise pós-lançamento recorrente para investir incrementalmente em melhorias ao longo do tempo com uma abordagem iterativa.

Uma prática recomendada é usar infrastructure-as-code modelos, como AWS CloudFormation o Terraform da Hashicorp, para a configuração de sua infraestrutura VPCs, como configurações de sub-rede e dependências necessárias para iniciar serviços essenciais de jogos, para que você possa consultar esses modelos, personalizá-los rapidamente, se necessário, e implantá-los em locais onde seja necessária infraestrutura adicional para oferecer suporte aos seus jogadores.

Você também deve se certificar de que entendeu como sua estratégia de implantação atual pode ser desenvolvida para permitir uma expansão futura. Os modelos de IaC são repetíveis, mas não substituem o planejamento de rede. O IPAM gerencia seu. VPCs Dimensionamento da sub-rede, seleção da zona de disponibilidade e alinhamento do inventário de IP e da zona de disponibilidade entre contas. É importante considerar a rede e pode causar interrupções para os jogadores quando alterada. Servidores de jogos implantados em várias localizações geográficas se conectarão ao back-end do jogo, que é mais comum estar hospedado em uma ou várias regiões de origem, o que pode exigir configuração adicional para oferecer suporte à conectividade privada. Essas considerações devem ser avaliadas continuamente ao longo do tempo para que você possa fazer alterações em sua estratégia de hospedagem de jogos à medida que os requisitos do jogo evoluem ou os requisitos do jogador mudam.

Ao determinar quantos locais de hospedagem de jogos usar para seu jogo, considere os seguintes fatores:

- Melhoria na qualidade da experiência do jogador: quanto da melhoria da experiência do jogador você pode introduzir adicionando mais locais de hospedagem de jogos? Qual é o ganho

incremental de desempenho que você pode obter ao fazer isso? Como você medirá essa melhoria de desempenho?

- Quais populações de jogadores priorizar: para quantos jogadores você pode melhorar a experiência se adicionar mais locais de hospedagem de jogos? Quais populações de jogadores ou localizações geográficas você priorizará?
- Impactos posteriores da mudança: Se você mudar sua estratégia de hospedagem de jogos, como isso influenciará seus tempos de espera de matchmaking para os jogadores? O tamanho das partidas, o equilíbrio de habilidades ou o número de jogadores no pool de jogadores podem acomodar uma mudança na estratégia de localização da hospedagem do jogo? Oferecer suporte a mais locais pode potencialmente fragmentar o pool de jogadores e aumentar o custo e a complexidade.

Cada uma dessas considerações deve ser avaliada à medida que você determina onde adicionar ou remover locais de hospedagem de jogos. Por exemplo, você pode optar por priorizar a melhoria da experiência para jogadores em localizações geográficas com a menor experiência de jogo ou para jogadores que expressam o maior feedback público. Você também pode optar por incluir a monetização do jogador em suas prioridades, por exemplo, concentrando a atenção em melhorar a experiência de jogadores em localizações geográficas que geram uma fonte significativa de receita para seu jogo ou têm o potencial de gerar receita incremental se você introduzir melhorias no desempenho.

Além de hospedar a infraestrutura em Regiões da AWS, você pode usar as Zonas Locais, que são uma extensão de uma Região da AWS, para hospedar seus servidores de jogos e outros aplicativos sensíveis à latência, como servidores de bate-papo por voz, mais próximos dos seus jogadores. Você também pode optar por executar a infraestrutura de desenvolvimento de jogos em Zonas Locais para melhorar a experiência de suas equipes de desenvolvimento de jogos. Por exemplo, você pode usar as Zonas Locais para tratar de casos de uso, como hospedar réplicas de seus servidores autogerenciados de controle de origem mais perto dos desenvolvedores de jogos e oferecer estações de trabalho virtuais e armazenamento de conteúdo para usuários que usam EC2 instâncias da Amazon, volumes do EBS e sistemas de FSx arquivos da Amazon implantados em uma ou mais Zonas Locais perto de seus estúdios de desenvolvimento sem exigir que você hospede a infraestrutura localmente.

Outposts são uma boa opção quando Regiões ou Zonas Locais não estão disponíveis na mesma área geográfica. A conectividade do seu data center AWS deve ser considerada para permitir que o servidor de jogos tenha a confiabilidade do sistema de back-end. AWS Outposts e os servidores Outpost foram criados especificamente para serem executados AWS em seu datacenter usando

os mesmos serviços e APIs para ajudar a criar um modelo de implantação consistente onde quer que você execute seu jogo. Vários racks podem ser combinados em um Posto Avançado lógico e a infraestrutura pode ser compartilhada entre eles. Contas da AWS O ciclo de vida do hardware é gerenciado por AWS e o prazo de entrega pode ser de apenas 3 meses.

Se você estiver criando jogos usando contêineres e quiser a flexibilidade de adotar uma arquitetura de implantação híbrida usando software de código aberto que pode ser implantado em sua própria infraestrutura local, você pode usar o ECS Anywhere ou o EKS Anywhere como alternativa às Zonas Locais. AWS Outposts Se você hospeda na Amazon GameLift, o Amazon GameLift Anywhere pode ser usado para executar seu servidor baseado em hardware local, o que pode acelerar seu processo de desenvolvimento, permitindo que você use Zonas Locais ou registre seu próprio metal como parte de sua frota.

Etapas de implementação

- Use infrastructure-as-code ferramentas como AWS CloudFormation o Terraform para implantações reproduzíveis, permitindo rápida personalização e escalabilidade dos locais de hospedagem de jogos com base nas necessidades do jogador.
- Avalie as melhorias na experiência do jogador, as prioridades da população de jogadores e os impactos posteriores, como tempos de matchmaking, ao adicionar ou remover locais de hospedagem de jogos.
- Use AWS Locais Zones, Outposts ou opções híbridas, como ECS Anywhere, EKS Anywhere ou GameLift Anywhere, para otimizar a infraestrutura sensível à latência e dar suporte a diversas necessidades de implantação.

Desenvolvimento iterativo

GAMEPERF03: Como você pode usar a Amazon GameLift para obter eficiência de desempenho de desenvolvimento iterativo?

GameLift A Amazon fornece um end-to-end fluxo de trabalho para desenvolvimento e teste de desempenho do seu jogo em um ambiente de teste local.

Práticas recomendadas

- GAMEPERF03-BP01 Use o Amazon GameLift Anywhere e um kit de ferramentas de teste GameLift
- GAMEPERF03-BP02 Teste o desempenho e a escalabilidade dos servidores de jogos
- GAMEPERF03-BP03 Otimize a utilização de recursos de contêineres GameLift

GAMEPERF03-BP01 Use o Amazon GameLift Anywhere e um kit de ferramentas de teste GameLift

Para melhorar a eficiência do desempenho por meio de um processo de desenvolvimento iterativo, utilize o Amazon GameLift Anywhere junto com o Amazon GameLift Testing Toolkit para estabelecer um ambiente de teste abrangente.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Essa abordagem permite iteração rápida, coleta eficiente de dados e análise detalhada do desempenho. As principais etapas incluem:

Crie um ambiente de teste

Use o Amazon GameLift Anywhere para configurar um ambiente de teste local ou baseado na nuvem. Essa configuração elimina a necessidade de enviar cada iteração de compilação do servidor de jogo para uma frota gerenciada, reduzindo o tempo de ativação.

Integre o Amazon GameLift Testing Toolkit

Incorpore o Amazon GameLift Testing Toolkit ao seu fluxo de trabalho de desenvolvimento. O kit de ferramentas fornece scripts, ferramentas e bibliotecas para visualizar a GameLift infraestrutura da Amazon, lançar players virtuais e iterar os conjuntos de FlexMatch regras com o simulador. FlexMatch Ele simplifica a integração e o gerenciamento dos GameLift recursos da Amazon, permitindo que você automatize tarefas comuns e reúna os dados necessários para a análise de desempenho.

Ciclos rápidos de construção e teste

Atualize rapidamente a frota de testes com novas versões, inicie-a e inicie os testes. Isso facilita um build-test-repeat ciclo rápido, permitindo que os desenvolvedores validem vários aspectos da experiência do jogador no jogo, incluindo interações multijogador.

Testes abrangentes

Teste a integração do servidor de jogos com o SDK GameLift do servidor Amazon, interações de serviços de back-end, configurações de matchmaking e outros recursos de hospedagem. GameLift Utilize o kit de ferramentas de GameLift teste para automatizar os testes e reunir métricas de desempenho detalhadas, garantindo que os componentes do jogo funcionem perfeitamente juntos.

Analise os dados de desempenho

Use os dados coletados pelo Kit de Ferramentas GameLift de Teste para analisar gargalos de desempenho e otimizar seu servidor de jogos. O kit de ferramentas ajuda a rastrear as principais métricas, identificar problemas e tomar decisões baseadas em dados para melhorar a eficiência do desempenho.

Ao incorporar o Amazon GameLift Anywhere e o kit de ferramentas de GameLift teste em seu processo de desenvolvimento iterativo, você pode melhorar significativamente a eficiência do desempenho por meio de testes rápidos, verificações de integração abrangentes e análises detalhadas de desempenho.

Etapas de implementação

- Use o Amazon GameLift Anywhere para criar um ambiente de teste, reduzindo o tempo de ativação para compilações de servidores de jogos e permitindo uma iteração rápida.
- Integre o Amazon GameLift Testing Toolkit para automatizar tarefas de teste, simular jogadores e FlexMatch validar configurações durante o desenvolvimento.
- Colete e analise dados de desempenho com o kit GameLift de ferramentas de teste para identificar gargalos, otimizar servidores de jogos e melhorar a eficiência do desempenho.

GAMEPERF03-BP02 Teste o desempenho e a escalabilidade dos servidores de jogos

Para testar o desempenho e a escalabilidade dos seus servidores de jogos, implemente uma estrutura de testes robusta usando os GameLift recursos da Amazon e o kit de ferramentas GameLift de teste.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

As principais práticas incluem:

Teste iterativo

Use uma frota do Amazon GameLift Anywhere para criar um ambiente hospedado na nuvem, onde você pode criar e testar componentes do jogo de forma iterativa. Esse ambiente deve refletir as condições de hospedagem do mundo real, permitindo testes realistas de desempenho e escalabilidade.

Teste de integração de servidores de jogos

Teste a integração do seu servidor de jogos com o SDK do GameLift servidor Amazon, incluindo o início de novas sessões de jogo e o rastreamento de eventos da sessão de jogo usando AWS CLI nosso kit de ferramentas GameLift de teste. Isso verifica se o servidor do jogo funciona corretamente no GameLift ambiente.

Use o kit GameLift de ferramentas de teste para automatizar os testes e coletar métricas de desempenho detalhadas. O kit de ferramentas permite que você visualize a GameLift infraestrutura, lance players virtuais para testes de carga e repita os conjuntos de FlexMatch regras com o simulador. FlexMatch É particularmente útil para escalar tarefas do ECS Fargate, que simulam sessões de jogadores criando várias sessões de jogo simultâneas para testar o estresse da infraestrutura do servidor.

Teste de escalabilidade

Experimente designs de filas de sessões de jogos, frotas com vários locais, frotas spot e sob demanda e vários tipos de instâncias. Teste as opções de posicionamento da sessão de jogo, as políticas de latência e as configurações de priorização da frota. Configure o escalonamento da capacidade para atender à demanda dos jogadores e valide se o sistema pode lidar com a carga esperada sob diferentes condições.

Etapas de implementação

- Use o Amazon GameLift Anywhere para configurar um ambiente de teste realista para testes iterativos de desempenho e escalabilidade.
- Teste a integração do servidor de jogos com o SDK GameLift do servidor, facilitando o gerenciamento correto da sessão e o rastreamento de eventos no GameLift ambiente.

- Realize testes de escalabilidade com o kit de ferramentas de GameLift teste, simulando a carga de jogadores, testando filas de sessões e validando o dimensionamento da frota, as políticas de latência e as configurações de priorização.

GAMEPERF03-BP03 Otimize a utilização de recursos de contêineres GameLift

Para otimizar a utilização dos recursos dos GameLift contêineres, projete sua frota de contêineres de forma eficaz e defina limites precisos de recursos.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

As principais diretrizes incluem:

- Design de grupos de contêineres: organize seu software em grupos de contêineres. O contêiner principal deve agrupar seu aplicativo de servidor de jogos e o Amazon GameLift Agent. Use contêineres auxiliares para obter software adicional para gerenciar dependências e definir limites específicos de contêineres para uso de memória e CPU.
- Defina limites de recursos: para cada grupo de contêineres, determine os recursos necessários de memória e CPU. Defina limites opcionais para contêineres individuais para verificar se eles têm recursos reservados, mas também podem exceder esses limites se houver recursos adicionais disponíveis. Isso ajuda a evitar a contenção de recursos e possíveis falhas no contêiner.
- Grupo de contêineres de daemons: considere usar um grupo de contêineres de daemons para processos de monitoramento ou em segundo plano que não precisem ser escalados com o grupo de contêineres primário. Isso verifica se as tarefas essenciais em segundo plano são realizadas de forma eficiente sem afetar os processos primários do servidor do jogo.

Etapas de implementação

- Crie grupos de contêineres com um contêiner primário para o servidor do jogo e o GameLift Agente, e sidecars para gerenciar dependências, com limites específicos de memória e CPU.
- Defina limites de recursos para cada grupo de contêineres para reservar os recursos necessários e, ao mesmo tempo, permitir o uso controlado dos recursos para evitar contenções.

- Use um grupo de contêineres daemon para tarefas em segundo plano ou de monitoramento, certificando-se de que eles operem com eficiência sem afetar os processos primários do servidor do jogo.

Computação e hardware

GAMEPERF04: Como você impede que as sessões de jogo afetem os jogadores que estão executando na mesma instância do servidor de jogos?

Depois que o servidor do jogo estiver sendo executado AWS, você precisará monitorar seu desempenho para fornecer uma experiência de qualidade ao jogador, independentemente da utilização de recursos, da computação subjacente ou da saturação.

Práticas recomendadas

- GAMEPERF04-BP01 Monitore os processos do servidor de jogos para detectar problemas
- GAMEPERF04-BP02 Teste o desempenho do seu servidor de jogos com cenários de jogo simulados e reais

GAMEPERF04-BP01 Monitore os processos do servidor de jogos para detectar problemas

Você pode executar vários processos de servidor de jogos por instância para utilizar com eficiência os recursos em suas instâncias de servidor de jogos. Nesse caso, projete sua arquitetura de forma que um processo individual do servidor de jogos que hospeda uma sessão de jogo não cause impacto adverso em outras sessões de jogo hospedadas na mesma instância. Use métricas para entender como o posicionamento do jogo e o tipo de modo de jogo podem afetar o desempenho das instâncias do servidor de jogos. Incorpore uma combinação de processos de baixa carga (lobby, loja ou tutorial para um jogador) e de alta carga (jogabilidade ranqueada, multijogador ou de alta habilidade) para evitar problemas na instância do servidor do jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Monitore a experiência do jogador por meio de métricas do lado do cliente e do servidor, coletando telemetria para tempo de ping e instabilidade, quedas de quadros, tempo de resposta da API, erros e conclusão bem-sucedida do loop de jogo. Correlacione os registros de data e hora desses eventos com problemas de suporte ao jogador e registros do servidor para identificar gargalos de desempenho. Ferramentas como Dtrace, ftrace, uperf e eBPF podem ser usadas para investigação e análise aprofundadas do desempenho do sistema.

Implemente o monitoramento dos recursos limitados disponíveis para suas instâncias de servidor de jogos para que você possa gerar alertas quando processos individuais do servidor de jogos estiverem ultrapassando limites predeterminados de orçamento de recursos. Quando os limites são violados, convém configurar o software do servidor de jogos para despejar os registros relevantes do sistema e do servidor de jogos em um armazenamento durável, como uma solução de registro central, para que os engenheiros do servidor de jogos possam investigar esse comportamento. Além disso, sua instância do servidor de jogos deve ser configurada para relatar métricas de cada um dos processos do servidor de jogos em execução na instância, para que você possa monitorar esses processos individuais do servidor de jogos, além das métricas gerais da instância do servidor de jogos.

Por exemplo, GameLift fornece métricas para monitorar sessões de jogos, que podem ser aumentadas com métricas e registros personalizados específicos do jogo coletados usando o Amazon CloudWatch Agent, que você pode configurar na sua instância do servidor de jogos. Suas métricas podem ser visualizadas CloudWatch ou exportadas para outras ferramentas, como o Amazon Managed Grafana, que é integrado ao Single Sign-On para facilitar o acesso às métricas por usuários que talvez não tenham acesso ao Management Console. Consulte as seguintes melhores práticas para gerenciar registros e métricas usando a Amazon GameLift, que também fornece suporte para visualizar registros de sessões de jogos individuais.

Etapas de implementação

- Execute vários processos de servidor de jogos por instância e combine modos de jogo de baixa e alta carga para evitar problemas e verificar a utilização equilibrada dos recursos.
- Monitore métricas do lado do cliente e do servidor, como ping, instabilidade, quedas de quadros e tempos de resposta da API, e correlacione-as com os registros do servidor e os problemas relatados pelos jogadores para identificar gargalos.

- Configure o monitoramento de recursos para cada processo do servidor de jogos, gere alertas para violações de limites e armazene registros em armazenamento durável para análise usando ferramentas como o CloudWatch Amazon Managed Grafana.

GAMEPERF04-BP02 Teste o desempenho do seu servidor de jogos com cenários de jogo simulados e reais

Realize testes de desempenho e avalie vários cenários de jogo para determinar se o processo do servidor de jogos lida adequadamente com a utilização de recursos fixos, como memória de EC2 instância, CPU e largura de banda de rede.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

A criação de testes de jogabilidade simulados com bots que podem espelhar caminhos e comportamentos comuns de seus jogadores pode determinar como os processos do servidor de jogos lidam com isso em diferentes cenários de uso. Por exemplo, você pode implementar uma solução, como o Distributed Load Testing, AWS que pode ser personalizada para executar simulações de clientes de jogos ou compilações de clientes de jogos para gerar cenários de jogo. Execute testes de jogo internos e use equipes de controle de qualidade para testar o estresse dos vários recursos do seu jogo, para que você possa desenvolver a confiança de que seu jogo foi projetado para ter um desempenho ideal. AWS Device Farmpode ser usado para realizar testes móveis e na web para seus jogos para iOS, Android e navegador em vários tipos de dispositivos.

Etapas de implementação

- Realize testes de desempenho com bots simulando comportamentos comuns de jogadores para avaliar a utilização de recursos do servidor de jogos em diferentes cenários.
- Use soluções como o Distributed Load Testing on AWS para personalizar e simular cenários de jogo para testes de estresse.
- Faça testes de jogo internos e use ferramentas como AWS Device Farm para testes de jogos em dispositivos móveis e navegadores em vários dispositivos.

Seleção de computação

GAMEPERF05: Como você seleciona a solução computacional apropriada para o seu jogo?

O desempenho computacional varia de acordo com os tamanhos e famílias de instâncias. É vantajoso usar várias opções de computação provenientes de pools de capacidade separados. Desenvolva uma estratégia de composição de frota que dê preferência ao desempenho, mas inclua diversidade suficiente para evitar erros de capacidade insuficientes.

Práticas recomendadas

- GAMEPERF05-BP01 Compare o desempenho do seu jogo em vários tipos de computação
- GAMEPERF05-BP02 Mova tarefas de computação para fluxos de trabalho assíncronos non-latency-sensitive

GAMEPERF05-BP01 Compare o desempenho do seu jogo em vários tipos de computação

Para cargas de trabalho de servidores de jogos, não há uma abordagem única para identificar a solução computacional ideal para hospedar seu servidor de jogos. Uma estratégia comum para comparar servidores de jogos é começar com instâncias EC2 'c' otimizadas para computação, porque essa família de instâncias fornece alto desempenho para cargas de trabalho que são computacionalmente intensivas. Como alternativa, se seu jogo exigir uma quantidade significativa de memória para implementar recursos específicos, as instâncias otimizadas para memória podem ser as mais adequadas.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Se sua carga de trabalho utiliza recursos de rede significativos, considere implementar instâncias otimizadas para a rede, o que normalmente é indicado usando um 'n' no nome da instância. Evite instâncias intermitentes do tipo 't', pois depois que os créditos se esgotarem, o desempenho diminuirá. Os jogos são sensíveis à latência e aos pacotes descartados, portanto, é recomendável usar uma rede EC2 aprimorada para ajudar a melhorar o desempenho da rede de seus servidores de jogos. A rede aprimorada usa a I/O virtualização de raiz única (SR-IOV) para fornecer recursos

de rede de alto desempenho em tipos de instância compatíveis. O SR-IOV é um método de virtualização de dispositivos que fornece maior I/O desempenho e menor utilização da CPU quando comparado às interfaces de rede virtualizadas tradicionais. A rede avançada fornece uma largura de banda maior, uma performance melhor de pacotes por segundo (PPS) e latências entre instâncias consistentemente mais baixas. A rede aprimorada com o Elastic Network Adapter está disponível para os tipos de EC2 instância mais recentes e é importante atualizá-la regularmente para se beneficiar dos aprimoramentos de desempenho de instâncias mais recentes e do AWS hipervisor Nitro.

Se seu jogo tiver um desempenho semelhante em vários tipos de EC2 instância, considere usar vários tipos de instância para hospedar seus servidores de jogos. Monitore o desempenho ao longo do tempo e realize mais otimizações depois de hospedar sessões de jogos de produção suficientes para poder identificar tendências de desempenho. Lembre-se de que seus requisitos de computação podem mudar à medida que você adiciona novos recursos ao jogo que exigem alocação diferente de recursos. Você pode configurar grupos de EC2 Auto Scaling para usar vários tipos de instância ou pode usar grupos de Auto Scaling separados para hospedar instâncias de servidores de jogos que executam tipos de instância separados, o que pode simplificar o gerenciamento da correlação e agregação de métricas.

Avalie o desempenho do seu jogo em diferentes tipos de processadores, como instâncias baseadas em Intel, instâncias baseadas em AMD e instâncias Graviton baseadas em ARM. O Unreal Engine 5.1.1 ou mais recente pode compilar servidores de jogos para Graviton e melhorar o preço/desempenho do seu jogo. Realize testes de varredura e saturação em vários tamanhos dentro de cada família para determinar o ponto ideal em que a utilização e o desempenho são consistentes.

Você também deve avaliar como o desempenho do seu jogo é afetado quando ele é hospedado usando contêineres e funções Lambda. Para casos de uso em que processos de servidor de jogos de longa duração não são necessários, como jogos assíncronos e serviços de back-end de jogos, considere usar uma arquitetura sem servidor com o Lambda, que pode simplificar o gerenciamento e as operações das equipes de operações de jogos, além de permitir que você implante seu jogo globalmente com mais rapidez para muitos. Regiões da AWS Para obter as melhores práticas sem servidor, consulte o Serverless Applications Lens - Well-Architected Framework.

Etapas de implementação

- Compare servidores de jogos em instâncias 'c' otimizadas para computação para cargas de trabalho com uso intenso de CPU, instâncias otimizadas para memória para tarefas com muita memória e instâncias 'n' otimizadas para rede para alta taxa de transferência de rede.

- Use redes aprimoradas com o Elastic Network Adapter (ENA) em instâncias compatíveis para melhorar o desempenho da rede, reduzir a latência e aumentar as taxas de processamento de pacotes.
- Avalie e teste vários tipos de instância, processadores (Intel, AMD, Graviton) e opções de hospedagem de contêineres ou Lambda, ajustando as soluções de computação à medida que os recursos do jogo evoluem.

Para obter mais informações, consulte Escolha a estratégia de computação certa para seus servidores globais de jogos.

GAMEPERF05-BP02 Mova tarefas de computação para fluxos de trabalho assíncronos non-latency-sensitive

Ao otimizar o desempenho do seu jogo, é importante ter em mente que somente algumas das interações entre o cliente e o back-end do jogo devem ser realizadas de forma síncrona. Você deve considerar cada recurso do ponto de vista da experiência do jogador e determinar se determinadas interações exigem comunicações síncronas, que são bloqueadoras e consomem muitos recursos, ou se esses recursos podem ser implementados de forma assíncrona. Ao implementar chamadas de rede, use uma abordagem assíncrona e sem bloqueio. Além disso, o back-end do jogo também deve ser configurado para realizar o trabalho de maneira eficiente, transferindo tarefas para filas e priorizando respostas rápidas aos clientes sempre que possível.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Por exemplo, a atualização de uma tabela de classificação no final de uma sessão de jogadores pode ser implementada de forma assíncrona para que o cliente não precise esperar que a atualização da tabela de classificação seja concluída. Em vez disso, implemente isso de forma assíncrona no cliente do jogo e considere projetar seu serviço de back-end para colocar esses tipos de operações em filas, como o Amazon SQS. Com essa arquitetura, configure seu back-end para aceitar a solicitação, enfileirá-la no SQS, o que ajuda a armazenar mensagens de forma durável para processamento assíncrono e responder prontamente ao cliente. Quando a atualização da tabela de classificação for concluída, o back-end poderá enviar uma atualização ao cliente do jogo para que a visão do jogador sobre a tabela de classificação seja atualizada.

Como alternativa, o jogador pode simplesmente visitar a tela da tabela de classificação do seu jogo para recuperar os dados mais recentes, o que pode emitir uma solicitação da web ao seu back-end para recuperar os dados mais recentes do cache.

Etapas de implementação

- Determine se as interações cliente-back-end exigem comunicação síncrona; implemente abordagens assíncronas e sem bloqueio sempre que possível para otimizar o uso de recursos.
- Use o Amazon SQS para liberar tarefas não críticas, como atualizações na tabela de classificação.
- Permita que o cliente busque dados atualizados de forma assíncrona, como recuperar os dados mais recentes da tabela de classificação sob demanda ou por meio de atualizações em segundo plano.

Recursos

- Entendendo o sistema de mensagens assíncronas para microsserviços
- Lambda — Usando integrações de serviços e processamento assíncrono

Gerenciamento de dados

GAMEPERF06: Como você gerencia e analisa com eficiência os registros do servidor de jogos e armazena diferentes tipos de dados do jogo para obter um desempenho ideal?

Os jogos podem ter dados do jogador, registros do jogo e registros do servidor, que devem ser separados uns dos outros sempre que possível. A centralização da ingestão de registros e do gerenciamento do ciclo de vida pode beneficiar suas equipes de jogo, fornecendo informações sobre o que está acontecendo no jogo e no servidor.

Práticas recomendadas

- GAMEPERF06-BP01 Centralize a coleta e o armazenamento de registros
- GAMEPERF06-BP02 Categorize e armazene dados do jogo com base nos padrões de acesso
- GAMEPERF06-BP03 Habilite a formatação e o agrupamento de registros eficientes
- GAMEPERF06-BP04 Implemente políticas de rotação e retenção de registros

- GAMEPERF06-BP05 Use ferramentas de monitoramento e visualização

GAMEPERF06-BP01 Centralize a coleta e o armazenamento de registros

Implemente uma solução centralizada de coleta e armazenamento de registros para coletar registros de instâncias do servidor de jogos e. GameLift

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Use serviços como o Amazon CloudWatch Logs para coletar, monitorar e armazenar dados de log de seus servidores e GameLift instâncias de jogos. CloudWatch O Logs fornece uma solução escalável e totalmente gerenciada para gerenciamento de registros, facilitando o armazenamento e a recuperação eficientes dos dados de log sem afetar o desempenho do servidor de jogos. Se você estiver executando o agente CloudWatch Logs, considere os vários tipos de instalação e opções de configuração, como tamanho do lote e duração do buffer, para minimizar o impacto no servidor do jogo. Considere as instâncias do servidor de jogos efêmeras e reduza a dependência do registro localizado sempre que possível. Estabeleça uma política centralizada para a implementação das melhores práticas de registro.

Etapas de implementação

- Use o Amazon CloudWatch Logs para coletar, monitorar e armazenar dados de log de instâncias de servidores de jogos e GameLift facilitar o gerenciamento de registros centralizado e escalável.

GAMEPERF06-BP02 Categorize e armazene dados do jogo com base nos padrões de acesso

Categorize os dados do seu jogo em diferentes tipos com base em seus padrões de acesso e requisitos de armazenamento.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

As categorias comuns incluem dados de jogadores, jogos salvos, armazenamento mundial persistente e dados analíticos.

Etapas de implementação

Use soluções de armazenamento apropriadas para cada tipo de dados para otimizar o desempenho e a economia:

- Dados do jogador: use o Amazon DynamoDB, um banco de dados NoSQL rápido e escalável, para armazenar perfis, preferências e dados de progressão dos jogadores. O acesso de baixa latência e os recursos de escalabilidade automática do DynamoDB oferecem recuperação e atualização eficientes dos dados do player.
- Jogos salvos: use o Amazon S3 para armazenar jogos salvos e pontos de verificação. O S3 oferece alta durabilidade e escalabilidade para armazenar grandes quantidades de dados salvos de jogos. Considere usar o S3 Transfer Acceleration ou o Amazon CloudFront para fazer uploads e downloads mais rápidos de jogos salvos.
- Armazenamento mundial persistente: para jogos com estados mundiais persistentes ou dados de jogos compartilhados, considere usar o Amazon DynamoDB, o Amazon ou o ElastiCache Amazon MemoryDB. ElastiCache e o MemoryDB fornecem armazenamento de valores-chave na memória, enquanto o DynamoDB é um banco de dados NoSQL baseado em SSD. Esses serviços fornecem acesso rápido aos dados armazenados, reduzindo o tempo necessário para que o processo do servidor de jogos salve o estado do jogo, o que melhora o desempenho geral do processo.
- Dados analíticos: use o Amazon Managed Streaming for Apache Kafka ou o Kinesis Data Streams para ingerir fluxos de dados de seus produtores de dados de jogos. O Amazon Managed Service para Apache Flink pode ser usado para transformação e análise em tempo real e enviado ao Amazon Data Firehose para processamento e entrega em data lakes de back-end, armazéns e serviços de análise. A orientação para o Game Analytics Pipeline em AWS ilustra como os serviços trabalham juntos para fornecer análises em lote e quase em tempo real.

GAMEPERF06-BP03 Habilite a formatação e o agrupamento de registros eficientes

Configure seus processos de servidor de jogos para gerar registros de forma estruturada e em um formato que possa ser analisado, como JSON.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Implemente técnicas de agrupamento de registros para minimizar a frequência das transferências de dados de registro dos servidores do jogo para o armazenamento de registros centralizado. Os registros em lote reduzem a sobrecarga da rede e melhoram o desempenho do servidor de jogos. Use registros detalhados ou de nível de depuração como uma exceção e não um padrão, pois eles podem incorrer em uma penalidade de desempenho e custo que deve ser evitada sempre que possível.

GAMEPERF06-BP04 Implemente políticas de rotação e retenção de registros

Estabeleça políticas de rotação e retenção de registros para gerenciar o crescimento dos dados de registro e otimizar a utilização do armazenamento.

Nível de risco exposto se esta prática recomendada não for estabelecida: Baixo

Orientação para implementação

Configure seus servidores de jogo para alternar automaticamente os registros com base no tamanho ou nos intervalos de tempo. Defina políticas de retenção de CloudWatch logs no Amazon Logs para arquivar ou excluir automaticamente dados de log antigos que não são mais necessários para análise ativa ou solução de problemas.

GAMEPERF06-BP05 Use ferramentas de monitoramento e visualização

Use ferramentas de monitoramento e visualização para obter informações sobre o desempenho do seu servidor de jogos e identificar oportunidades de otimização.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Use CloudWatch a Amazon para monitorar as principais métricas e configurar alarmes para notificações proativas. Utilize ferramentas como o Amazon Managed Service for Prometheus e o Amazon Managed Grafana para coletar, consultar e visualizar métricas de seus servidores e infraestrutura de jogos. Crie painéis informativos para monitorar o desempenho, identificar gargalos e fazer otimizações baseadas em dados.

Rede e entrega de conteúdo

GAMEPERF07: Como você projeta seu serviço de matchmaking para otimizar o desempenho?

É importante considerar a habilidade do jogador, a qualidade do provedor de serviços de Internet (ISP) e a distribuição da população de jogadores como uma dimensão do ajuste de desempenho. As sessões de jogo podem ser colocadas em servidores estrategicamente localizados para nivelar o campo de jogo e sediar um jogo justo.

Práticas recomendadas

- GAMEPERF07-BP01 Defina limites de latência de rede para seu jogo
- GAMEPERF07-BP02 Execute um serviço de matchmaking separado para cada modo de jogo e região de hospedagem de jogos
- GAMEPERF07-BP03 Monitore regularmente o desempenho de matchmaking
- GAMEPERF07-BP04 Monitore regularmente o desempenho da rede
- GAMEPERF07-BP05 Use a tecnologia de aceleração de rede para melhorar o desempenho na Internet

GAMEPERF07-BP01 Defina limites de latência de rede para seu jogo

Ao desenvolver um jogo multijogador, verifique se a infraestrutura do jogo não introduz latência desnecessária para os jogadores. Se o seu jogo for sensível à latência da rede, você deve definir limites de latência em sua lógica de matchmaking para priorizar a colocação de jogadores em sessões de servidores de jogos hospedadas em servidores de jogos próximos ou Regiões da AWS que atendam ao seu objetivo de proporcionar uma experiência de jogador ideal.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Em muitos jogos sensíveis à latência, é comum instrumentar os clientes do jogo para fazer ping em cada um dos locais de infraestrutura do jogo para coletar dados de desempenho, como latência da rede, instabilidade e perda de pacotes, e reportar esses dados ao back-end de coleta de métricas para que possam ser analisados. Ao combinar jogadores em sessões de jogo, você pode configurar

seu jogo para incorporar a latência de rede percebida pelo cliente do jogo à sua infraestrutura de servidor de jogo como uma das entradas usadas na lógica do serviço de matchmaking.

GAMEPERF07-BP02 Execute um serviço de matchmaking separado para cada modo de jogo e região de hospedagem de jogos

Se seu jogo oferece vários modos de jogo para os jogadores escolherem, você deve separar os sistemas de matchmaking de cada um deles para poder ajustar de forma independente o desempenho de cada modo de jogo com base em seus requisitos exclusivos e reduzir a disputa de recursos. Cada modo de jogo provavelmente terá requisitos exclusivos de latência aceitável, tamanho da partida e outras lógicas de combinação personalizadas específicas do jogo. Eles provavelmente também atrairão diferentes tipos de jogadores. Execute o serviço de matchmaking de cada modo de jogo como uma implantação de software separada para que você possa testar o desempenho e operar os modos de jogo de forma independente.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Por exemplo, você pode executá-las como funções Lambda separadas para cada modo de jogo ou como implantações separadas de serviços baseados em contêineres.

Implante seus serviços de matchmaking em várias regiões próximas às localizações dos seus servidores de jogo. O tráfego de jogadores seguirá várias rotas, por isso é importante que o serviço de matchmaking mantenha um perfil de up-to-date latência em várias ISPs para melhorar a eficiência do posicionamento de sessões de jogo de baixa latência. GameLift FlexMatch fornece orientação adicional para selecionar regiões para matchmakers, e inclui a capacidade de integrar seus matchmakers com filas de sessões de jogos em várias regiões.

GAMEPERF07-BP03 Monitore regularmente o desempenho de matchmaking

Uma das maneiras mais notáveis de otimizar o desempenho de um jogo para os jogadores é reduzir o tempo que eles precisam esperar antes de entrarem em uma sessão de jogo. Longos tempos de espera podem fazer com que os jogadores percam o interesse e causar desgaste, por isso é importante considerar isso ao projetar sua solução de matchmaking.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao projetar sua configuração de matchmaking para o seu jogo, crie regras que determinem as condições aplicadas para formar uma partida. Você deve considerar o impacto que essas regras terão no desempenho do sistema, principalmente nos tempos de espera dos jogadores. Antes de implantar mudanças em sua implementação de matchmaking, como a adição de novas condições ou filtros de matchmaking, teste isso com antecedência ou considere liberar essa alteração gradualmente para uma pequena amostra de jogadores como um canário ou A/B teste para coletar métricas de desempenho antes de introduzir essa mudança em toda a população de jogadores.

Configure seu serviço de matchmaking para gerar registros detalhados para entender as condições ou regras que foram aplicadas a cada solicitação de matchmaking. Isso auxilia na revisão e no ajuste da implementação do matchmaking conforme necessário.

Por exemplo, GameLift FlexMatch A Amazon fornece um serviço de matchmaking totalmente gerenciado que pode ser usado como um serviço independente com sua própria hospedagem de servidor de jogos ou usado com servidores de jogos hospedados na Amazon. GameLift FlexMatch pode gerar notificações de eventos para a Amazon EventBridge, consulte Configurar notificações de FlexMatch eventos. Use o Amazon Simple Notification Service (Amazon SNS) para receber dados de matchmaking no formato JSON, permitindo que você processe e armazene automaticamente essas informações para análise a fim de melhorar o desempenho do matchmaking.

Configure métricas para monitorar quanto tempo seu serviço de matchmaking leva para encontrar uma sessão de jogo adequada para os jogadores. Revise as métricas de duração do matchmaking regularmente e correlacione esses horários com o comportamento do jogador e o sentimento da comunidade. Use esses dados para desenvolver limites adequados para tempos limite de matchmaking que podem ser incluídos na configuração da regra de matchmaking.

Por exemplo, a Amazon GameLift FlexMatch fornece suporte para definir tempos limite de solicitação de matchmaking, bem como criar regras de matchmaking que podem permitir que os requisitos diminuam com o tempo. Esse recurso permite que você crie matchmaking que pode se adaptar para facilitar a criação de partidas e colocar os jogadores em sessões de jogo quando as partidas são difíceis de encontrar.

GAMEPERF07-BP04 Monitore regularmente o desempenho da rede

Para jogos competitivos, é importante ter uma experiência de jogador consistente.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Um jogo que dura 50 ms de forma confiável para uma base maior de jogadores é mais justo e divertido do que uma partida em que um jogador tem 10 ms de ping e outro tem 70 ms de ping. Mudanças no roteamento do ISP podem afetar parte da população de jogadores, e seu sistema de matchmaking precisará se adaptar. O Amazon CloudWatch Network Monitoring ajuda a determinar se o problema está no seu jogo ou no provedor de internet do jogador.

Etapas de implementação

- Use o Amazon Cloudwatch Network Monitoring para monitorar o desempenho da rede e identificar problemas de roteamento.
- Use os registros de fluxo da VPC para identificar padrões de tráfego anormais ou pacotes descartados, o que pode indicar congestionamento da rede, problemas de ISP ou configurações incorretas que afetam a latência do player.

GAMEPERF07-BP05 Use a tecnologia de aceleração de rede para melhorar o desempenho na Internet

Além de colocar fisicamente a infraestrutura de jogo sensível à latência mais próxima dos jogadores, você também pode melhorar a experiência do jogador otimizando o desempenho da rede do seu jogo. AWS usa o protocolo BGP para influenciar o roteamento da Internet para usar o caminho mais rápido dos provedores de serviços de Internet para nossa rede de fronteira. Se você opera sua própria rede e precisa de mais controle e observabilidade sobre o comportamento de roteamento e a publicidade do BGP, você pode usar o peering privado ou rotear o tráfego da Internet Direct Connect para o jogo em execução. AWS

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

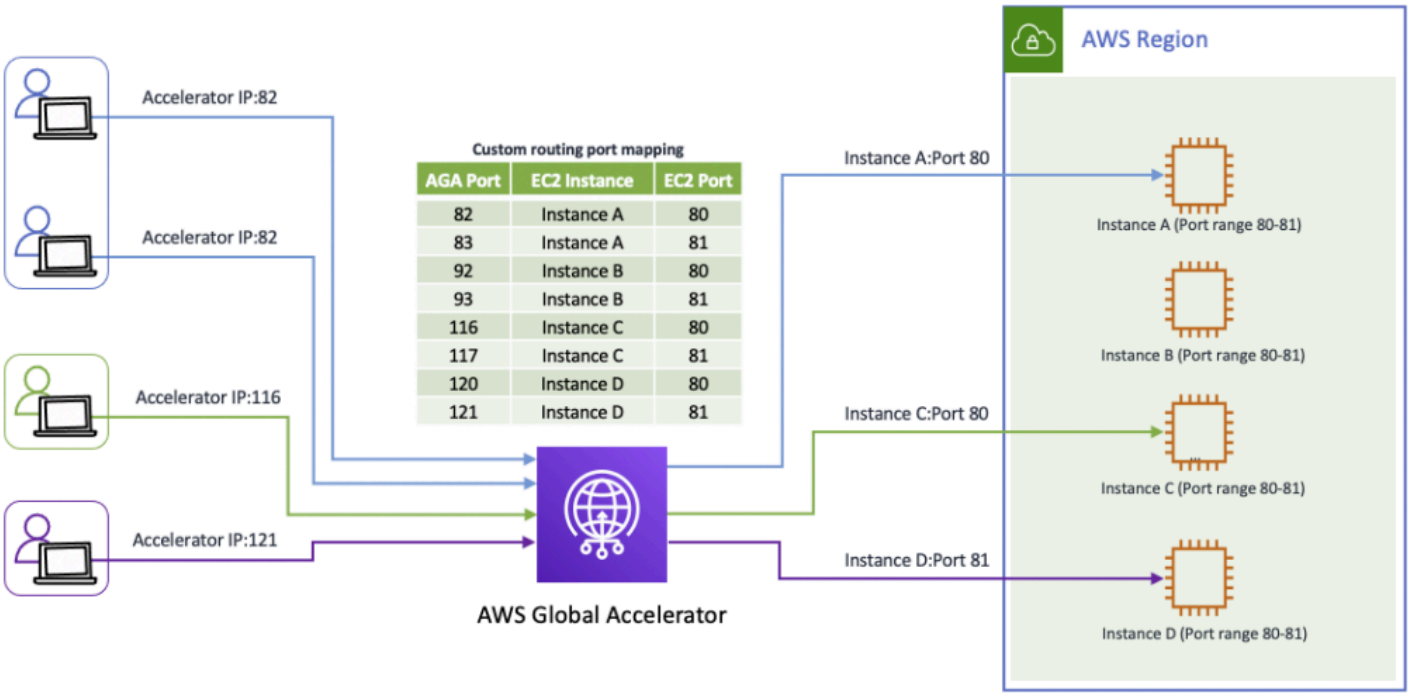
Orientação para implementação

Considere a arquitetura de referência a seguir para oferecer suporte à melhoria do desempenho e da capacidade de resposta da Internet.

Desempenho de rede aprimorado para jogos usando o Global Accelerator

Para uma solução totalmente gerenciada para roteamento de rede, o AWS Global Accelerator melhora o desempenho da rede do seu aplicativo usando a rede AWS global, que pode ser usada

para acelerar o tráfego de jogos, o bate-papo por voz e o tráfego de mensagens em tempo real, bem como outros aplicativos sensíveis à latência, ao mesmo tempo em que fornece um failover rápido para seus servidores de jogos. Os aceleradores de roteamento personalizados do Global Accelerator podem ser integrados ao seu serviço de matchmaking para fornecer roteamento determinístico de vários jogadores para a mesma sessão de jogo usando portas e endereços IP anycast estáticos.



Suas equipes de desenvolvimento de jogos podem estar distribuídas em todo o mundo e exigir acesso eficiente a conteúdo ou ativos compartilhados. Para melhorar o desempenho do conteúdo compartilhado armazenado em buckets do Amazon S3, você pode configurar a replicação bidirecional de seus dados entre regiões usando a replicação entre regiões do S3 para que os usuários possam acessar dados de buckets mais próximos a eles. Para simplificar esse padrão de acesso, use os pontos de acesso multirregionais do S3, que aceleram as solicitações para o S3 pela rede global usando o Global Accelerator.

Para obter mais informações, consulte Melhorando a experiência do jogador usando o AWS Global Accelerator e o Amazon GameLift FleetiQ.

Etapas de implementação

- Use o AWS Global Accelerator para ajudar a melhorar o desempenho da rede para tráfego de jogos, bate-papo por voz e mensagens em tempo real, ao mesmo tempo em que facilita o failover rápido para servidores de jogos.

- Configure os aceleradores de roteamento personalizados do Global Accelerator para se integrarem ao seu serviço de matchmaking, permitindo o roteamento determinístico dos jogadores para as sessões de jogo usando anycast estático. IPs
- Ative a replicação entre regiões do S3 para replicar o conteúdo compartilhado entre regiões para equipes distribuídas de desenvolvimento de jogos.
- Use os pontos de acesso multirregionais do S3 para acelerar o acesso aos dados do S3 na rede AWS global para usuários distribuídos globalmente.

Processo e cultura

GAMEPERF08: Como você alinha a barra de desempenho do seu jogo às expectativas dos jogadores e desenvolvedores?

Conhecer seu player e desenvolvedor é um dos aspectos mais importantes para melhorar a eficiência do desempenho. Oferecer um jogo de alto desempenho com baixa sobrecarga operacional é uma das melhores maneiras de mostrar aos jogadores e desenvolvedores que você se preocupa com a experiência deles e pode diferenciar seu jogo e seu estúdio.

Práticas recomendadas

- GAMEPERF08-BP01 Informe e inclua o jogador em seu processo
- GAMEPERF08-BP02 Alinhe a seleção de soluções com as habilidades e a experiência da equipe de engenharia

GAMEPERF08-BP01 Informe e inclua o jogador em seu processo

Forneça uma opção para exibir métricas no jogo, como latência, quadros por segundo e pacotes descartados. Supere problemas de infraestrutura e tempo de inatividade para manutenção por meio de comunicações voltadas para o jogador, como páginas de status. Comemore novos locais de jogo com comunicações de jogadores, incluindo blogs de desenvolvedores, e defina as expectativas de melhorias esperadas na experiência do jogador.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Inclua o player

Forneça um processo simples de envio de diagnóstico que colete arquivos relevantes e os anexe a um ticket de suporte ao jogador do seu cliente de jogo. Habilite um fórum de suporte onde os jogadores possam ajudar uns aos outros e se tornar parte da melhoria da experiência de jogo

Considere as compensações versus as expectativas do jogador

Mover sistemas de back-end para obter eficiência de custos pode não ser perceptível para os jogadores, mas mover servidores de jogos pode alterar o tempo de ping. Seja consistente e justo com os jogadores com os motivos para expandir e reduzir seus locais de hospedagem de jogos.

As comunidades e regiões geográficas de jogadores terão suas próprias características que podem afetar as expectativas do seu jogo. Por exemplo, a Coreia do Sul tem a Internet mais rápida do planeta e a expectativa de jogabilidade é de latência de um dígito, o que impulsiona o jogo altamente competitivo. A jogabilidade casual em dispositivos móveis cria um perfil de desempenho e um padrão de uso diferentes em comparação com o jogo em sessão de console e PC.

O login e o lobby fazem parte da experiência e devem ser responsivos, mesmo que o servidor esteja off-line para manutenção. Planejar uma noite de invasão ou passear no saguão faz parte da experiência do jogador e é importante considerar ao escolher áreas de foco para eficiência de desempenho. Os jogadores podem deixar seu cliente de jogo aberto por meses; às vezes, eles podem fazer login ocasionalmente para ler as notas do patch. Um jogo Live Ops precisa ter em mente toda a experiência do jogador como parte do processo e da cultura de engenharia.

Etapas de implementação

- Forneça métricas no jogo, como latência, FPS e perda de pacotes, e comunique problemas de infraestrutura e cronogramas de manutenção por meio de páginas de status e atualizações voltadas para o jogador.
- Implemente um recurso de diagnóstico de despejo e envio no cliente do jogo e crie um fórum de suporte para promover a solução de problemas e a melhoria orientadas pela comunidade.
- Adapte as otimizações de desempenho às expectativas da comunidade de jogadores, como baixa latência para regiões competitivas ou login/lobby experiências responsivas para jogadores casuais e de longa sessão.
- Crie fluxos de trabalho do Live Ops para considerar toda a experiência do jogador, desde a jogabilidade ativa até o comportamento inativo do cliente, facilitando o engajamento contínuo.

GAMEPERF08-BP02 Alinhe a seleção de soluções com as habilidades e a experiência da equipe de engenharia

Avalie as habilidades e a experiência da sua equipe em gerenciar e otimizar o desempenho do servidor de jogos ao escolher sua opção de hospedagem. Soluções auto-hospedadas, como contêineres EC2 e contêineres, exigem mais conhecimento sobre gerenciamento de infraestrutura, ajuste de desempenho e escalabilidade. Se sua equipe não tiver essas habilidades, um serviço gerenciado como esse GameLift pode ser mais adequado, pois elimina muitas das complexidades e permite que sua equipe se concentre em otimizações específicas do jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Ao avaliar esses fatores e realizar testes de desempenho em diferentes opções de hospedagem, você pode selecionar a solução mais adequada que atenda aos requisitos específicos do seu jogo e, ao mesmo tempo, otimizar a eficiência do desempenho.

Recursos

Saiba mais sobre nossas melhores práticas relacionadas à eficiência de desempenho.

Documentos relacionados:

- Centro de Arquitetura da AWS
- Pilar de eficiência de desempenho — AWS Well-Architected Framework
- Comparando seus padrões de armazenamento local com os serviços AWS de armazenamento
- Armazenamento de instâncias Armazenamento temporário em blocos para EC2 instâncias
- Colete métricas, registros e rastreamentos usando o CloudWatch agente
- CloudWatchAgente
- Como faço para ativar e configurar a rede aprimorada em minhas EC2 instâncias?
- Melhorando a experiência do jogador com o uso do AWS Global Accelerator e do Amazon FleetiQ GameLift
- Blog de tecnologia da Riot Games: testes de escalabilidade e carga para Valorant
- Jogos online em grande escala com uma solução híbrida AWS
- Método de saturação e erros de utilização (USE)

- Instâncias Amazon EC2 Spot
- Desempenho em grande escala com a Amazon ElastiCache
- Estratégias de cache de banco de dados usando Redis
- Opções de conectividade da Amazon Virtual Private Cloud
- Padrões de design de melhores práticas: otimizando o desempenho do Amazon S3

Benchmarks relacionados:

- Compare os volumes do Amazon EBS
- Largura de banda de rede da instância Amazon EC@

Ferramentas relacionadas:

- Unreal Engine: testando e otimizando seu conteúdo
- Criador de perfil Unity
- Sistema de qualidade Open 3D Engine (O3DE)
- Monitoramento de GameLift servidores Amazon
- Kit de ferramentas GameLift de teste da Amazon

Vídeos relacionados:

- AWS re:Invent 2019: [REPETIÇÃO 2] EC2 Fundações da Amazon (-R2) CMP211
- AWS re:Invent 2019: Potencializando a próxima geração da EC2 Amazon: um mergulho profundo no sistema Nitro (03-R2) CMP3
- Introdução ao Amazon GameLift FleetIQ — palestras técnicas on-line AWS
- Zach Blitz, da Riot Games, fala sobre como usar AWS para melhorar os jogos
- AWS re:Invent 2023 - AWS Graviton: a melhor relação preço/desempenho para suas cargas de trabalho () AWS CMP334

Treinamento relacionado:

- Hospedagem de servidores de jogos em AWS
- Usando o Amazon GameLiftFleet IQ para servidores de jogos

- Começando com AWS for Games — Parte I
- Hospedagem de servidores de jogos em AWS
- AWS re:Invent 2023 — AWS Graviton: a melhor relação preço/desempenho para suas cargas de trabalho () AWS CMP334

Otimização de custos

O pilar de otimização de custos inclui o processo contínuo de refinamento e aprimoramento de um sistema durante todo o seu ciclo de vida. Esse processo abrange desde o design inicial de sua primeira prova de conceito até a operação contínua das cargas de trabalho de produção. Ao adotar as práticas descritas neste paper, você pode criar e operar sistemas com reconhecimento de custos que alcancem os resultados comerciais desejados com o menor preço. A implementação dessas práticas de otimização de custos pode permitir que sua empresa maximize o valor do seu investimento em nuvem.

Os jogos são projetos criativos exclusivos que devem competir pela atenção e pelo tempo de jogo do jogador. Antes do lançamento, os desenvolvedores de jogos geralmente não têm uma compreensão clara de quão popular ou duradouro seu jogo se tornará. Dependendo da estratégia de monetização, das prioridades de negócios e do estágio do ciclo de vida do jogo, os desenvolvedores precisarão fazer concessões ao avaliar as decisões de otimização de custos.

Por exemplo, durante a fase de pré-lançamento de um novo jogo muito aguardado, o foco geralmente está no speed-to-market desenvolvimento de recursos e no desempenho. A prioridade é verificar se a infraestrutura pode ser dimensionada para atender ao pico de demanda dos jogadores. Por outro lado, se um jogo não for bem-sucedido ou o desenvolvimento estiver desacelerando, o foco pode mudar para reduzir ao máximo os custos para continuar operando o jogo para os jogadores existentes.

Muitos desenvolvedores de jogos também operam vários jogos simultaneamente, o que requer considerações adicionais. Recursos como infraestrutura, software e equipe podem ser compartilhados em vários jogos ao vivo, permitindo que as perdas de um jogo sejam compensadas pelos lucros de outro. Nesse cenário, o foco na otimização de custos pode melhorar as finanças em todo o portfólio de jogos.

Considerando os modelos de negócios exclusivos, a escala e a imprevisibilidade dos jogos, as seguintes perguntas-chave podem orientar as decisões de otimização de custos:

- Como posso medir o custo da infraestrutura por jogador, sistema e recurso do jogo?
- Qual é o equilíbrio certo entre otimização de custos e experiência do jogador no estágio atual do ciclo de vida do meu jogo?
- Como posso maximizar o retorno sobre o investimento usando o modelo de precificação certo para meus AWS recursos?

Aplicar essas melhores práticas e fazer as perguntas certas pode ajudar os desenvolvedores de jogos a criar e operar sistemas com reconhecimento de custos que alcancem resultados comerciais e minimizem os custos.

Áreas de foco

Princípios de design

Além dos princípios de design do pilar de otimização de custos do Well-Architected Framework, os princípios de design a seguir otimizam os custos de execução da carga de trabalho do jogo na nuvem.

- Meça o custo da infraestrutura por jogador, sistema e recurso do jogo: entenda e acompanhe os custos de infraestrutura necessários para experiências e recursos específicos dos jogadores em todos os sistemas de jogo. Isso pode identificar áreas de sua arquitetura que podem precisar de otimização de custos.
- Avalie a relação entre otimização de custos e experiência do jogador: avalie em que estágio seu jogo se encontra para determinar o foco certo: experiência do jogador ou otimização de custos. Normalmente, quando um jogo atinge a massa crítica e a população de jogadores se estabiliza, é hora de focar na otimização dos custos operacionais. A chave é equilibrar a oferta de uma ótima experiência ao jogador com a administração da infraestrutura de jogos da maneira mais econômica. A implementação desses princípios de design maximiza o retorno sobre seus investimentos em jogos.

Gerenciamento financeiro na nuvem

Não há práticas recomendadas de gerenciamento financeiro em nuvem específicas para o Games Lens. Consulte o pilar de otimização de custos do Well-Architected Framework para obter orientação sobre gerenciamento financeiro na nuvem.

Reconhecimento de despesas e usos

GAMECOST01: Como você mede o custo de seus ambientes de jogo?

Entenda o custo por jogador, o recurso do jogo e o ambiente para que você possa gerenciar e prever seus gastos à medida que o número de jogadores muda com o tempo e os recursos são adicionados e aprimorados. Considere as práticas recomendadas a seguir para gerenciar os custos dos diferentes ambientes de jogo.

Práticas recomendadas

- GAMECOST01-BP01 Implemente a atribuição de custo por jogador, recurso do jogo e ambiente
- GAMECOST01-BP02 Descubra oportunidades de otimização

GAMECOST01-BP01 Implemente a atribuição de custo por jogador, recurso do jogo e ambiente

A atribuição de custos para servidores de jogos geralmente é mais simples de executar do que serviços de back-end de jogos, porque um servidor de jogos geralmente é otimizado para hospedar um número específico de jogadores simultâneos por instância, que pode ser amortizado pelo custo de execução da instância.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Para serviços de back-end de jogos, é recomendável separar os componentes do seu jogo em recursos distintos que podem ser gerenciados como recursos lógicos ou físicos separados para facilitar a análise de custos.

Por exemplo, embora possa parecer simples implementar um único aplicativo monolítico para hospedar serviços de back-end de jogos, esse padrão dificulta a obtenção do custo total por jogador e por recurso do jogo ao longo do tempo à medida que você adiciona mais recursos, pois os custos de computação, rede e armazenamento dos recursos são compartilhados entre os recursos. Considere adotar uma arquitetura sem servidor para seus serviços de back-end de jogos com serviços como Amazon API Gateway AWS Lambda e/ou AWS Fargate para computação, Amazon SQS e Amazon SNS para mensagens, Amazon S3 para armazenamento de objetos e Amazon DynamoDB para armazenamento de banco de dados. Esses serviços são apenas alguns exemplos de produtos que oferecem preços baseados no uso e orientados principalmente pelo volume de solicitações, para que os custos possam ser visualizados com granularidade. Recursos individuais, como funções Lambda, serviços Fargate, tabelas do DynamoDB e buckets do S3, podem ser associados a tags de alocação de custos para que você possa atribuir os custos desses serviços a nomes de recursos do jogo que facilitem a compreensão dos custos de cada um dos seus serviços.

Também é recomendável gerenciar separadamente cada um dos seus ambientes de desenvolvimento de jogos para que você possa atribuir custos aos diferentes ambientes. Normalmente, os desenvolvedores de jogos gerenciam ambientes separados para ambientes de desenvolvimento, teste, preparação e produção, conforme descrito no pilar de operações dessa lente da indústria de jogos. Cada ambiente geralmente tem requisitos diferentes de escalabilidade, desempenho e uso e pode ser gerenciado por equipes separadas. Para controlar os custos, organize esses ambientes para que você possa monitorar e atribuir adequadamente os custos de cada ambiente.

Para obter mais informações, consulte a documentação a seguir:

- Criando um jogo multijogador sem servidor que se expande
- Servidores de sessão de jogo autônomos com um back-end WebSockets baseado
- Servidores de sessão de jogo autônomos com um back-end sem servidor

Etapas de implementação

- Separe os serviços de back-end de jogos em recursos distintos usando arquiteturas sem servidor ou em contêineres, como o Amazon AWS Lambda API Gateway, AWS Fargate e para permitir a atribuição granular de custos por recurso.
- Aplique tags de alocação de custos a recursos individuais (por exemplo, funções Lambda, tabelas do DynamoDB e buckets do S3) para associar custos a recursos específicos do jogo para uma melhor análise de custos.

- Gerencie ambientes separados para desenvolvimento, teste, preparação e produção, organizando e monitorando seus custos de forma independente para se alinhar aos requisitos de escalabilidade e uso.

GAMECOST01-BP02 Descubra oportunidades de otimização

Desenvolvedores e editores de jogos podem usar AWS FinOps práticas para ajudar a otimizar seus custos de nuvem e obter melhor visibilidade de seus gastos com a nuvem. Ao fazer isso, os produtores de jogos podem alinhar o custo médio necessário para manter a infraestrutura dos jogadores com os resultados financeiros fornecidos pelo jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Baixo

Orientação para implementação

AWS oferece uma orientação de solução pronta para uso para o Cloud Financial Management para gerenciar e otimizar suas despesas com serviços em nuvem. Esse recurso inclui visibilidade granular e análise de custo e uso para apoiar a tomada de decisões em tópicos como painéis de gastos, otimização, limites de gastos, reembolso e detecção e resposta de anomalias. A orientação da solução para o Cloud Financial Management inclui recursos de orçamento e previsão, oferecendo uma arquitetura definida e otimizada para suas cargas de trabalho, para que você possa selecionar o modelo de preços certo e atribuir custos de recursos relevantes às suas equipes. Isso ativa técnicas de rastreamento, notificação e otimização de custos em seu ambiente e recursos. Você pode gerenciar centralmente as informações de despesas e dar acesso às partes interessadas essenciais, conforme necessário, para obter visibilidade direcionada e apoiar a tomada de decisões.

Outra FinOps ferramenta importante é o Cost Optimization Hub, que fornece uma visão centralizada das recomendações e oportunidades de otimização de custos em todo o seu Contas da AWS país Regiões da AWS, para que você possa aproveitar ao máximo seus AWS gastos. Você pode usar o Cost Optimization Hub para identificar, filtrar e agregar recomendações de otimização de AWS custos em seu Contas da AWS e. Regiões da AWS Ele faz recomendações sobre o dimensionamento correto de recursos, a exclusão de recursos ociosos, Savings Plans e Instâncias Reservadas. Com um único painel, você evita a necessidade de acessar vários AWS produtos para identificar oportunidades de otimização de custos.

Se suas equipes de jogos estiverem usando Contas da AWS o MyApplications in Console de gerenciamento da AWS Home compartilhado, pode ser usado para visualizar os custos de recursos do aplicativo para cargas de trabalho individuais. Essa visão granular permite identificar as

tendências de custo específicas em sua infraestrutura de jogos, permitindo que você tome decisões informadas sobre alocação e otimização de recursos.

Além disso, revisar regularmente seus dados de faturamento e gerenciamento de custos com o AWS Data Exports revela oportunidades ocultas de economia de custos. Esse relatório detalhado fornece uma análise abrangente de seus gastos com a nuvem, permitindo que você identifique áreas de gastos excessivos, recursos não utilizados e oportunidades de aproveitar serviços ou modelos de preços mais econômicos.

Ao adotar FinOps princípios e aproveitar as ferramentas fornecidas por AWS, os desenvolvedores e editores de jogos podem fazer o uso mais eficiente de seus recursos de nuvem, melhorando seus resultados financeiros e liberando fundos para mais desenvolvimento e inovação de jogos.

Etapas de implementação

- Use ferramentas de gerenciamento Nuvem AWS financeiro para obter visibilidade granular e detalhada, painéis de gastos, detecção de anomalias e atribuição de custos para otimizar e monitorar as despesas na nuvem de forma eficaz.
- Use o Cost Optimization Hub para centralizar as recomendações de dimensionamento correto, Savings Plans e Instâncias Reservadas em todas as regiões. Contas da AWS
- Analise regularmente os dados AWS de faturamento usando Exportações de Dados e outros MyApplication AWS para ajudar a analisar os custos específicos da carga de trabalho, descobrir oportunidades de economia e otimizar a alocação de recursos.

Recursos economicamente eficientes

GAMECOST02: Como você está escolhendo a solução de computação certa para seus servidores de jogos?

Um dos aspectos mais exclusivos da carga de trabalho de um jogo, em comparação com outros tipos de cargas de trabalho, é o servidor do jogo. O servidor do jogo é fundamental para a experiência do jogador, pois os jogadores se conectam a ele a partir do cliente do jogo para jogar uma sessão de jogo.

O servidor do jogo também é um dos maiores fatores de custo para operar um jogo multijogador. Portanto, é importante otimizar a forma como você utiliza a infraestrutura computacional dos seus servidores de jogos para reduzir custos.

Práticas recomendadas

- GAMECOST02-BP01 Otimize o custo da transferência de dados pela Internet
- GAMECOST02-BP02 Otimize o número de sessões de jogo hospedadas em cada instância do servidor de jogos para otimizar custos
- GAMECOST02-BP03 Selecione a opção de preço de computação apropriada para reduzir custos

GAMECOST02-BP01 Otimize o custo da transferência de dados pela Internet

Embora cobrem AWS principalmente pela transferência de dados de saída (saída) de seus AWS recursos para a Internet, as empresas de jogos podem enfrentar altos custos relacionados à transferência de dados por meio dos balanceadores de carga AWS Direct Connect ou AWS Gateway, que podem cobrar pelos dados de entrada (entrada) e saída. Implemente soluções que reduzam o custo geral de transferência de dados do AWS back-end do jogo para seus jogadores, concentrando-se em minimizar as cobranças de saída de seus AWS recursos, bem como avaliar as opções para gerenciar as taxas de entrada e saída por meio de serviços de conectividade. AWS

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Use CloudFront a Amazon para reduzir o custo de entrega de conteúdo e aplicativos web voltados para o público.

O conteúdo e os ativos do jogo armazenados na nuvem geralmente são armazenados no Amazon S3 e entregues ao cliente do jogo diretamente do S3 ou de servidores web hospedados na Amazon EC2 que recuperam o conteúdo do Amazon S3 e o entregam aos clientes. Para reduzir os custos de transferência de dados dos downloads de conteúdo, considere usar a Amazon CloudFront em frente ao seu armazenamento em nuvem para entregar conteúdo aos usuários.

O uso CloudFront pode reduzir o custo da transferência de dados, pois custa menos entregar seu conteúdo CloudFront points-of-presence do que diretamente das regiões e CloudFront não cobra taxas de recuperação de origem para origens AWS baseadas, como Amazon EC2 e Amazon S3.

Se seu conteúdo for estático e não for alterado com frequência, você pode usá-lo CloudFront para armazenar esses dados em cache mais perto dos usuários finais, o que pode reduzir ainda mais os custos.

CloudFront também melhora a eficiência de custos de aplicativos e serviços web voltados para o público, mesmo que o armazenamento em cache não seja usado, pois o custo da transferência de dados entre seus servidores e clientes pode ser reduzido roteando o tráfego pela rede. AWS

A Amazon CloudWatch pode ser usada para monitorar seu CloudFront uso da Amazon. Para casos de uso em que você usa várias redes de entrega de conteúdo (CDNs), o Amazon CloudFront Origin Shield pode fornecer uma camada adicional de armazenamento em cache para consolidar e reduzir o número de solicitações de origem de diferentes provedores.

Para entender o tráfego da rede do seu jogo, você pode habilitar o VPC Flow Logs e o Amazon CloudWatch Internet Monitor para ter end-to-end visibilidade nas conexões de back-end do jogador ou do jogo. Essa abordagem pode identificar as causas do alto custo de transferência de dados e realizar mudanças na arquitetura para otimizar os gastos com transferência de dados.

Etapas de implementação

- Use a Amazon CloudFront em frente ao Amazon S3 ou às origens de conteúdo EC2 baseadas para reduzir os custos de transferência de dados, aproveitando a entrega de menor custo CloudFront points-of-presence e removendo as taxas de recuperação de origem.
- Habilite o VPC Flow Logs e o Amazon CloudWatch Internet Monitor para analisar o tráfego de rede e identificar mudanças na arquitetura para otimizar os custos de transferência de dados.
- Implemente o CloudFront Origin Shield para consolidar e reduzir as solicitações de origem ao usar várias CDNs para aumentar a eficiência de custos.

Para obter mais práticas recomendadas para entrega de conteúdo, consulte o whitepaper Entrega de conteúdo para jogos.

GAMECOST02-BP02 Otimize o número de sessões de jogo hospedadas em cada instância do servidor de jogos para otimizar custos

Otimize o número de sessões de jogo hospedadas por instância do servidor para obter uma melhor utilização da computação e reduzir os custos da infraestrutura computacional.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Para otimizar os custos, os desenvolvedores de jogos devem maximizar o número de sessões de jogo hospedadas no mesmo servidor físico ou virtual, também conhecido como densidade de empacotamento de seus servidores de jogos. Isso é obtido aumentando o número de processos do servidor de jogos que podem ser hospedados simultaneamente em uma instância.

Normalmente, um único processo de servidor de jogo não deve exigir o uso de todos os recursos disponíveis na EC2 instância. Essa é uma das formas mais importantes de reduzir os custos de computação de um jogo e requer o uso de um software que possa gerar e gerenciar vários processos do servidor na EC2 instância em portas separadas.

Por exemplo, a Amazon GameLift tem uma cota no número máximo de processos de servidor de jogos por instância, que você deve se esforçar para utilizar para reduzir os custos de hospedagem. Para obter mais informações, consulte GameLift os endpoints e cotas da Amazon Servers para obter detalhes sobre a cota atual para o máximo de processos do servidor de jogos por instância.

Como alternativa à implantação de processos de servidor de jogos em máquinas virtuais, como EC2 instâncias, está se tornando popular que os desenvolvedores de jogos executem seus servidores de jogos como aplicativos baseados em contêineres usando soluções de orquestração de contêineres. Os desenvolvedores de jogos podem usar o Amazon Elastic Container Service (Amazon ECS) ou o Guidance for Game Server Hosting usando Agones e Open Match no Amazon EKS. Outra opção é o Game Server Hosting on AWS Fargate, um mecanismo de computação sem servidor que funciona com ECS e EKS, permitindo que você se concentre no jogo sem precisar gerenciar a infraestrutura subjacente.

As soluções de contêiner fornecem uma funcionalidade de agendamento de tarefas que pode encontrar automaticamente uma instância de contêiner disponível no cluster para hospedar seu contêiner de servidor de jogos com base nos requisitos de recursos e em outras lógicas de posicionamento especificadas por você. No entanto, é importante considerar como você gerenciará o comportamento de escalonamento e posicionamento dos jogadores de uma forma que não interrompa as sessões ativas dos jogadores.

Etapas de implementação

- Aumente a densidade de empacotamento executando vários processos de servidor de jogos por EC2 instância usando portas e software de gerenciamento de processos separados.
- Use soluções da Amazon GameLift ou de contêineres, como ECS, EKS ou, AWS Fargate para gerenciar os processos do servidor de jogos com eficiência e reduzir os custos de infraestrutura.

- Monitore continuamente a utilização de recursos para refinar a densidade da embalagem e manter a eficiência de custos sem comprometer a experiência do jogador.

GAMECOST02-BP03 Selecione a opção de preço de computação apropriada para reduzir custos

Execute testes de desempenho do software do servidor de jogos em vários tipos de instância e opções de computação para determinar qual opção é mais econômica para seu jogo.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Além de utilizar com eficiência os tipos de EC2 instância certos para sua carga de trabalho, considere quais das opções de preços de computação disponíveis são mais adequadas para suas metas de otimização de custos. Há várias opções de preços disponíveis, incluindo instâncias sob demanda, instâncias spot, instâncias reservadas e Savings Plans.

Os Savings Plans (SPs) oferecem descontos para computação ao assumir compromissos de uso e são ideais para cenários em que você não consegue prever o uso esperado por um período de 1 ou 3 anos. Eles oferecem descontos como instâncias reservadas com a flexibilidade de aplicar esses descontos em todas as regiões, família de instâncias, sistema operacional e locação. Eles também podem ser aplicados a AWS Fargate quem pode ser uma opção de hospedagem de servidor de jogos para jogos casuais ou AWS Lambda quem é usado como uma ótima opção para jogos baseados em turnos que não exigem servidores de jogos. Para obter mais informações, consulte Criação de um jogo multijogador sem servidor que seja escalável.

Os Savings Plans são introduzidos durante o lançamento do jogo para reduzir os custos das cargas de trabalho dos servidores de jogos que estão contribuindo para o gasto de EC2 instâncias quando o jogo é lançado para o público. Os Savings Plans também podem ser introduzidos após o lançamento, quando a equipe de operações do jogo tem uma melhor compreensão do tráfego de jogadores após o jogo estar em produção por um longo período.

Como os Savings Plans oferecem flexibilidade regional, eles são particularmente ideais para otimizar os gastos com servidores de jogos para jogos com uso imprevisível em todas as regiões.

Por exemplo, se seu padrão diário de uso de jogadores exigir pelo menos 20 servidores para suportar sua base de jogadores, mas exigir periodicamente até 40 servidores, considere comprar os

compromissos do Savings Plan para cobrir a linha de base dos 20 servidores, pois essa demanda de uso é previsível e consistente e resultará na máxima utilização do compromisso de uso que você adquiriu.

Maximize a utilização dos Savings Plans e aumente-os com outras opções de compra que oferecem mais flexibilidade para picos imprevisíveis de uso do servidor de jogos, como instâncias sob demanda e spot, para obter economias ideais.

As instâncias spot são ideais para executar servidores de jogos porque oferecem os maiores descontos em computação, não exigem compromissos de uso e fornecem flexibilidade para tipos de carga de trabalho imprevisíveis e com picos. No entanto, as Instâncias Spot podem ser interrompidas, então elas são mais adequadas para cargas de trabalho de servidores de jogos com curta duração de sessão de jogo ou situações em que a tolerância à interrupção é maior.

Para obter mais informações sobre orientação para executar servidores de jogos usando o Kubernetes no Amazon EKS com instâncias EC2 spot, consulte Como executar jogos multijogador massivos com o Spot EC2 usando o Aurora Serverless.

Use as Instâncias EC2 Spot da Amazon para determinar grupos com a menor chance de interrupção que proporcionarão o máximo de economia em comparação com as tarifas sob demanda.

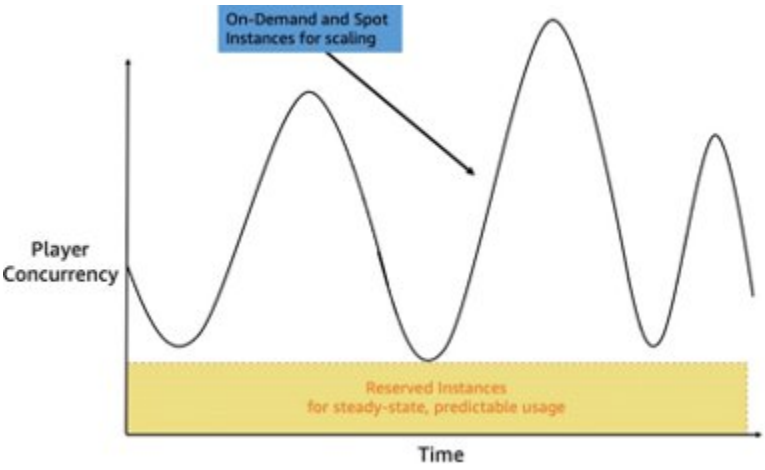
Ao usar o Spot, também é recomendável executar cargas de trabalho do servidor de jogos em vários tipos de EC2 instância e zonas de disponibilidade Região da AWS para diversificar o uso da capacidade e reduzir o risco de interrupção.

Considere o uso de instâncias spot em combinação com instâncias sob demanda para minimizar o impacto de possíveis interrupções nas sessões ativas do jogo e use a estratégia de alocação de capacidade otimizada para reduzir ainda mais o risco de interrupção.

Consulte as melhores práticas do Amazon EC2 Spot para obter melhores práticas adicionais. O rebalanceamento de capacidade no Auto Scaling para substituir instâncias spot em risco pode ser usado para monitorar proativamente e adicionar capacidade adicional quando as instâncias spot correm maior risco de interrupção.

O Amazon GameLift FleetiQ se integra às instâncias spot para otimizar o uso de instâncias spot de baixo custo e, ao mesmo tempo, reduzir o risco de interrupções. Se você estiver hospedando seu jogo usando GameLift, consulte a GameLift documentação para escolher os recursos de computação. Para obter mais informações, consulte Escolher recursos de computação para uma frota gerenciada.

O diagrama a seguir fornece um exemplo para ilustrar o uso de várias opções de preços de computação para cargas de trabalho de servidores de jogos:



Hospedagem de servidores de jogos com várias opções EC2 de preços

No diagrama, a simultaneidade dos jogadores flutua com o tempo, o que dificulta o gerenciamento da utilização e a otimização de custos. Para lidar com essa flutuação, considere adotar uma combinação de diferentes opções de preços de computação, usando Savings Plans para atender EC2 às necessidades de seus requisitos mínimos de uso e, ao mesmo tempo, contando com instâncias EC2 sob demanda e EC2 spot para atender às necessidades da demanda de seus jogadores.

Etapas de implementação

- Use Savings Plans para um uso básico previsível, combinando-os com instâncias spot e sob demanda para flexibilidade e otimização de custos durante picos de uso.
- Use instâncias spot para servidores de jogos com sessões curtas ou maior tolerância a interrupções, diversificando os tipos de instância e zonas de disponibilidade para minimizar o risco.
- Implemente ferramentas como o EC2 Spot Instances Advisor, o Capacity Rebalancing e o GameLift FleetiQ para otimizar o uso da Instância Spot e gerenciar proativamente as interrupções.

Custos de transferência de dados

GAMECOST03: Como você está otimizando os custos de transferência de dados para sua infraestrutura de jogos?

Os jogos podem transferir uma quantidade significativa de dados pela Internet entre os dispositivos clientes de jogos dos seus jogadores e sua infraestrutura de jogo para fornecer a experiência de jogo, bem como entre os componentes da sua infraestrutura de jogo.

Por exemplo, a transferência de dados ocorre quando os jogadores baixam atualizações de conteúdo do jogo para seus clientes de jogos, salvam o estado do progresso do jogo na nuvem, participam de sessões de jogo multijogador em tempo real com seus amigos e quando sua infraestrutura de jogo transfere dados entre regiões e zonas de disponibilidade. É importante entender onde ocorre a transferência de dados na carga de trabalho do jogo para otimizar suas opções de arquitetura e reduzir esse custo de transferência de dados.

Para otimizar os custos de transferência de dados para sua carga de trabalho de jogos, considere as seguintes práticas recomendadas:

Práticas recomendadas

- GAMECOST03-BP01 Escolha o tipo apropriado de armazenamento para conteúdo gerado pelo usuário para reduzir custos
- GAMECOST03-BP02 Otimize bancos de dados para back-ends de jogos

GAMECOST03-BP01 Escolha o tipo apropriado de armazenamento para conteúdo gerado pelo usuário para reduzir custos

Cada tipo de dado gerado e armazenado em seu jogo tem características únicas que você deve considerar ao determinar a solução de armazenamento certa para sua carga de trabalho.

Nível de risco exposto se esta prática recomendada não for estabelecida: Baixo

Orientação para implementação

Use o Amazon S3 Object Lifecycle Management para armazenar dados de objetos na categoria de armazenamento mais econômica. O Amazon S3 fornece várias classes de armazenamento e gerenciamento do ciclo de vida de objetos para facilitar a configuração de políticas simples e refinadas para fazer a transição automática de dados entre níveis de armazenamento para reduzir custos. Em vez de simplesmente armazenar dados na classe de armazenamento padrão S3 por padrão, considere definir uma configuração de ciclo de vida para fazer a transição automática de dados entre níveis ao longo do tempo, ou use a classe de armazenamento S3 Intelligent-Tiering para padrões de acesso desconhecidos ou variáveis.

Como alternativa, o S3 Intelligent-Tiering pode fazer a transição econômica e automática de dados entre níveis e é recomendado como uma classe de armazenamento padrão, pois fornece otimização de custos sem a necessidade de configurar manualmente políticas de ciclo de vida e agora é a melhor opção para objetos pequenos e de curta duração. Para obter mais informações, consulte Amazon S3 Intelligent-Tiering — Otimizações aprimoradas de custos para objetos pequenos e de vida curta.

Os casos de uso comuns do Amazon S3 incluem armazenamento de ativos de jogos, conteúdo estático, registros de jogos, armazenamento em data lake e backups. Para casos de uso em que sistemas de arquivos são necessários, como anexar sistemas de arquivos compartilhados às estações de trabalho durante o desenvolvimento, considere usar o Amazon Elastic File System (Amazon EFS), que fornece diferentes classes de armazenamento e aumenta e diminui automaticamente à medida que você adiciona e remove arquivos sem a necessidade de gerenciar a infraestrutura.

O Amazon S3 One Zone -IA é uma opção de armazenamento ideal para dados transitórios relacionados a sessões no jogo, matchmaking ou outras informações efêmeras que podem ser recriadas conforme necessário. Esse tipo de dados do jogo não requer redundância em várias zonas de disponibilidade ()AZs. Essa classe de armazenamento de baixo custo é adequada para registros de ações de jogadores, eventos de jogos e outros dados de telemetria usados para análise ou depuração.

O principal benefício de otimização de custos de usar o S3 Express One Zone para esses dados de jogos é a economia significativa em comparação com a classe de armazenamento S3 padrão, com uma redução de até 20% nos custos de armazenamento. Isso pode ser particularmente vantajoso para jogos com grandes volumes de dados que não exigem o mesmo nível de durabilidade e disponibilidade dos dados de aplicativos essenciais. Ao aproveitar o S3 One Zone, desenvolvedores e editores de jogos podem otimizar seus custos de armazenamento em nuvem sem comprometer a experiência geral do jogador.

Etapas de implementação

- Configure as políticas de ciclo de vida do Amazon S3 para fazer a transição de dados entre as classes de armazenamento ou use o S3 Intelligent-Tiering como padrão para otimização automática de custos com mudanças nos padrões de acesso.
- Use o S3 One Zone-Infrequent Access para dados transitórios da sessão de jogo, como registros de telemetria e matchmaking, para reduzir os custos de armazenamento em até 20% e, ao mesmo tempo, manter disponibilidade suficiente.

- Para necessidades de sistemas de arquivos compartilhados durante o desenvolvimento, use o Amazon EFS para simplificar o gerenciamento de armazenamento com capacidade elástica e várias classes de armazenamento.

GAMECOST03-BP02 Otimize bancos de dados para back-ends de jogos

Os jogos dependem muito de bancos de dados para armazenar uma ampla variedade de dados críticos, desde perfis e inventários de jogadores até microtransações e métricas de progressão no jogo. Os bancos de dados também desempenham um papel crucial no gerenciamento dos aspectos sociais dos jogos, como criar e manter grupos de jogadores, festas e aplicar políticas de moderação. À medida que a base de jogadores de um jogo cresce, os custos associados ao banco de dados aumentarão inevitavelmente para acomodar as crescentes demandas de dados e uso.

Nível de risco exposto se esta prática recomendada não for estabelecida: Médio

Orientação para implementação

Para back-ends de jogos executados no Amazon Aurora, há várias estratégias de otimização de custos que podem ser empregadas. Uma recomendação importante é escalar automaticamente suas réplicas de leitura com base nos padrões de uso, aumentando ou diminuindo dinamicamente o número de réplicas para lidar com as flutuações no tráfego. Isso significa que você está pagando pelos recursos de que realmente precisa. Outra tática de otimização é substituir as réplicas de leitura usadas para análise de jogos por exportações de DB snapshots para o Amazon S3, já que o serviço de armazenamento S3 geralmente é mais acessível do que as instâncias de banco de dados Aurora provisionadas. Para obter mais informações, consulte Exportação de dados de DB snapshots para o Amazon S3 para o Amazon RDS.

Explorar o uso de instâncias de banco de dados reservadas para o Amazon Aurora para suas principais instâncias de banco de dados e fazer a transição para a configuração Aurora Serverless também pode levar a economias substanciais de longo prazo, fornecendo mais flexibilidade e controle granular sobre a utilização de seus recursos.

Da mesma forma, para back-ends de jogos que usam o Amazon DynamoDB, empregar o modo de capacidade sob demanda do DynamoDB pode ser uma opção eficaz, especialmente para cargas de trabalho novas ou imprevisíveis, pois permite que você pague somente pelos recursos consumidos sem a necessidade de provisionamento excessivo. À medida que os padrões de tráfego do jogo se tornam mais estáveis e previsíveis com o tempo, você pode fazer a transição para o modo de capacidade provisionada do DynamoDB, que pode oferecer economia de custos por meio de um

melhor planejamento de capacidade. Ativar o auto-scaling em suas tabelas do DynamoDB é outra otimização fundamental, permitindo que o serviço ajuste dinamicamente a capacidade provisionada com base nas flutuações no tráfego. Teste a estrutura de dados do seu jogo em um ambiente de desenvolvimento antes do lançamento para encontrar e remover índices secundários locais (LSIs) e índices secundários globais (GSIs) desnecessários. Isso pode levar a uma economia substancial de custos no armazenamento e nas operações de dados do jogo. Remover operações de digitalização ineficientes do código de back-end do jogo em favor de consultas mais direcionadas, comprar capacidade reservada do Amazon DynamoDB e aproveitar os Streams do DynamoDB com AWS Lambda gatilhos para processar eventos de back-end do jogo pode otimizar ainda mais seus custos do DynamoDB. Para obter mais informações, consulte Práticas recomendadas para consultar e escanear dados no DynamoDB.

Ao implementar essas estratégias de otimização de custos para o Amazon Aurora e o DynamoDB, os desenvolvedores e editores de jogos podem reduzir significativamente seus gastos com bancos de dados de back-end de jogos.

Etapas de implementação

- Use o auto-scaling de réplicas de leitura do Aurora e as exportações de DB snapshots para o Amazon S3 para lidar com economia de tráfego e necessidades de análise flutuantes.
- Otimize os custos do DynamoDB começando com a capacidade sob demanda para novas cargas de trabalho, fazendo a transição para a capacidade provisionada com auto-scaling para tráfego previsível e removendo os não utilizados e. LSIs GSIs
- Evite operações de digitalização ineficientes em favor de consultas direcionadas, use instâncias reservadas ou capacidade reservada e use o AWS Lambda DynamoDB Streams com para processamento de eventos.

Gerenciando recursos de demanda e fornecimento

Não há melhores práticas de gerenciamento de recursos de oferta e demanda específicas para o Games Lens.

Para obter mais informações sobre como gerenciar a demanda e fornecer recursos, consulte Cost Optimization Pillar - Well-Architected AWS Framework.

Otimizando ao longo do tempo

Não há práticas recomendadas de otimização ao longo do tempo específicas para o Games Lens.

Para obter mais informações sobre como otimizar custos ao longo do tempo, consulte Cost Optimization Pillar - Well-Architected AWS Framework.

Recursos

Consulte os seguintes recursos para saber mais sobre as melhores práticas para otimização de custos:

Documentos relacionados:

- Como faço para reduzir as taxas de transferência de dados do meu gateway NAT na Amazon VPC?
- Apresentando o adaptador Amazon GameLift FleetiQ para Agones
- Como faço para encontrar os principais contribuidores para o tráfego do gateway NAT na minha Amazon VPC?
- Escolha a estratégia de computação certa para seus servidores de jogos globais
- AWS Well-Architected Labs — Recursos econômicos
- O plug-in CNI da Amazon VPC aumenta os limites de pods por nó
- Melhores práticas de arquitetura para otimização de custos
- Reduzindo o tempo de espera do jogador e dimensionando corretamente a alocação de computação usando o Amazon SageMaker AI RL e o Amazon EKS
- AWS Compute Optimizer
- A Electronic Arts otimiza os custos e as operações de armazenamento usando o Amazon S3 Intelligent-Tiering e o Amazon Glacier
- Evite práticas de licenciamento hostis migrando cargas de trabalho do Windows para o Linux
- Visão geral dos custos de transferência de dados para arquiteturas comuns
- AWS e a Kubecost colaboram para fornecer monitoramento de custos para clientes da EKS
- Estabelecendo a base: configurando seu ambiente para otimização de custos
- Visão geral das instâncias EC2 spot da Amazon

Sustentabilidade

O pilar de sustentabilidade fornece princípios de design, orientação operacional, melhores práticas e planos de melhoria para ajudar a atingir as metas de sustentabilidade para suas AWS cargas de trabalho.

Você pode encontrar orientações de implementação sobre implementação no whitepaper Sustainability Pillar - AWS Well-Architected Framework.

Áreas de foco

- Princípios de design
- Seleção da região
- Alinhamento com a demanda
- Software e arquitetura
- Gerenciamento de dados
- Hardware e serviços
- Recursos

Princípios de design

A sustentabilidade da indústria de jogos está evoluindo em taxas diversas em diferentes regiões do mundo. Com energia sustentável em grandes áreas da América do Norte e mandatos de sustentabilidade na UE e no Reino Unido, os arquitetos têm várias abordagens para alcançar cargas de trabalho mais ecológicas no futuro próximo. O pilar de sustentabilidade Well-Architected pode ser usado para alcançar essas medidas para uma ampla variedade de cargas de trabalho.

Esta seção da lente descreve várias práticas recomendadas que podem ser usadas para cargas de trabalho de jogos.

- Selecione o tipo de armazenamento apropriado para os dados do usuário de jogos.
- Esteja ciente das políticas de ciclo de vida de dados que eliminarão a duplicação de dados e eliminarão dados desnecessários de suas cargas de trabalho.
- Seja seletivo quanto ao dimensionamento correto da implantação de recursos computacionais.
- Use a tecnologia sem servidor para processos breves e transacionais.

Seleção da região

Não há melhores práticas de seleção de regiões específicas para o Games Lens. Para obter mais detalhes, consulte Sustainability Pillar - AWS Well-Architected Framework.

Alinhamento com a demanda

Não há alinhamento para exigir as melhores práticas específicas para o Games Lens. Para obter mais detalhes, consulte Sustainability Pillar - AWS Well-Architected Framework.

Software e arquitetura

Não há práticas recomendadas de software e arquitetura específicas para o Games Lens. Para obter mais detalhes, consulte Sustainability Pillar - AWS Well-Architected Framework.

Gerenciamento de dados

GAMESUS01: Como você gerencia os dados do usuário e do jogo no seu sistema de jogo?

Desenvolva uma estratégia de ciclo de vida de dados que otimize o armazenamento e a relevância dos dados ao reter apenas dados históricos críticos.

Práticas recomendadas

- GAMESUS01-BP01 Use tecnologias de armazenamento que se ajustem aos padrões adaptados ao conteúdo do usuário, às informações do assinante e às compras no jogo
- GAMESUS01-BP02 Use políticas de ciclo de vida ou expiração de TTL para excluir jogos, dados de usuários, arquivos de log ou ativos obsoletos desnecessários

GAMESUS01-BP01 Use tecnologias de armazenamento que se ajustem aos padrões adaptados ao conteúdo do usuário, às informações do assinante e às compras no jogo

Você deve classificar seus dados por tipo, necessidade de retenção e frequência de acesso. Isso permite que você selecione a solução de armazenamento mais otimizada para os inúmeros tipos

de dados que seu jogo ou serviços de back-end produzem. Os dados que mudam rapidamente devem ser armazenados em serviços de banco de dados de valores-chave ou na memória. Os dados transacionais devem ser armazenados em serviços de banco de dados relacional. Arquivos grandes, ativos de jogos ou conteúdo gerado pelo usuário devem ser armazenados em serviços de armazenamento de objetos.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Os jogos produzem e consomem uma grande variedade de tipos de dados que exigem soluções de armazenamento otimizadas para frequência de acesso, latência e custo. Os dados armazenados devem ser classificados usando tags para diferenciar os dados que podem ser removidos ou precisam ser armazenados a longo prazo.

Os serviços a seguir funcionam bem para uma variedade de casos de uso de jogos:

O Amazon Aurora (compatível com MySQL e PostgreSQL) oferece alta disponibilidade, baixa latência e escalabilidade automática, o que o torna uma excelente opção para lidar com grandes quantidades de dados transacionais, como gerenciamento e autenticação de contas de jogadores, economias no jogo, tabelas de classificação e classificações de jogadores, persistência do estado do jogo, gerenciamento de eventos e campanhas e implantação multirregional e de alta disponibilidade.

O Amazon DynamoDB é um banco de dados NoSQL totalmente gerenciado conhecido por sua baixa latência, alta taxa de transferência e escalabilidade perfeita, o que o torna ideal para lidar com dados de jogadores em tempo real, gerenciamento de sessões, inventário, economia no jogo, estado do jogo multijogador em tempo real, matchmaking, registro de eventos e escalabilidade para o público global.

O Amazon DocumentDB (compatível com o MongoDB) fornece um serviço de banco de dados escalável e de baixa latência orientado a documentos, perfeito para armazenar dados flexíveis e semiestruturados, como sistema de inventário, perfis e personalizações de jogadores, mundos de jogos e conteúdo gerado processualmente, interações sociais e de jogadores, análises e rastreamento de comportamento, além de metadados e configurações no jogo.

A Amazon ElastiCache oferece suporte ao armazenamento em cache na memória com Redis ou Memcached, oferecendo acesso rápido aos dados e tempos de resposta reduzidos, o que é fundamental para jogos multijogador em tempo real, nos quais velocidade e desempenho são essenciais para uma experiência de usuário tranquila. ElastiCache é utilizado em jogos para tabelas de classificação em tempo real, gerenciamento de sessões, armazenamento em cache de

metadados do jogo, bate-papo e mensagens no jogo, matchmaking, análise e telemetria em tempo real e escalonamento para eventos de alto tráfego.

O Amazon Simple Storage Service (S3) pode ser usado para armazenar objetos como ativos de jogos, vídeos, imagens, arquivos de log de texto e muito mais. O S3 é um serviço de armazenamento de objetos que oferece escalabilidade, disponibilidade de dados, segurança e desempenho líderes do setor.

Se ele oferece várias classes de armazenamento que oferecem suporte ao acesso frequente e pouco frequente aos dados e ao armazenamento de arquivos econômico. Para dados que são acessados com frequência durante o desenvolvimento, os estúdios devem armazenar objetos no S3 Standard para obter baixa latência e alto desempenho de taxa de transferência. Para dados que frequentemente vão de quentes a frios ou vice-versa, os estúdios devem investigar o S3 Intelligent-Tiering. O Intelligent-Tiering monitora os padrões de acesso de seus dados e move automaticamente os dados para o nível de acesso mais econômico.

Para estúdios que precisam de alta taxa de transferência, baixa latência e aceitam viver em uma única zona de disponibilidade, use o S3 Express One Zone. Isso replica os dados em uma única AZ e pode melhorar as velocidades de acesso aos dados em comparação com o padrão S3. Para necessidades de arquivamento profundo de dados históricos, a Amazon também oferece o Amazon Glacier. As classes de armazenamento do Amazon Glacier foram criadas especificamente para arquivamento de dados, oferecendo alto desempenho, flexibilidade de recuperação e armazenamento de arquivos de baixo custo na nuvem.

O Amazon Elastic Block Store pode ser usado para armazenar binários, arquivos executáveis e configurações dos servidores de jogos que seus servidores de jogos ou repositórios de ativos precisam para funcionar. Você deve capturar e excluir volumes não utilizados que não estejam anexados a uma EC2 instância. Isso alivia as despesas de armazenamento incorridas e, ao mesmo tempo, reduz o uso de serviços e hardware desnecessários.

Etapas de implementação

- Classifique os dados do jogo por tipo, necessidades de retenção e frequência de acesso, marcando os dados para distinguir entre requisitos de armazenamento de curto e longo prazo.
- Use o Amazon Aurora para dados transacionais, o DynamoDB para dados de jogadores em tempo real, o DocumentDB para dados semiestruturados e para armazenamento em cache de baixa latência de informações urgentes do jogo. ElastiCache
- Armazene ativos de jogos, registros e conteúdo gerado pelo usuário no Amazon S3, selecionando as classes de armazenamento apropriadas (por exemplo, Intelligent-Tiering, One Zone e Glacier)

com base nos padrões de acesso e nas necessidades de arquivamento, e use o EBS para binários e configurações de servidores de jogos com gerenciamento regular de snapshots.

GAMESUS01-BP02 Use políticas de ciclo de vida ou expiração de TTL para excluir jogos, dados de usuários, arquivos de log ou ativos obsoletos desnecessários

Você pode usar tags e tipos de dados para criar políticas de ciclo de vida ou TTLs para mover dados para o armazenamento de arquivos ou removê-los completamente do serviço. Isso pode incluir configurações temporárias, conteúdo arquivado expirado e registros históricos que não são mais necessários. A maioria dos serviços oferece suporte à marcação.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Para dados armazenados no S3, você pode usar políticas de ciclo de vida para mover os dados para níveis de armazenamento de acesso e arquivamento pouco frequentes. Em uma configuração do S3 Lifecycle, você pode definir regras para fazer a transição de objetos de uma classe de armazenamento para outra a fim de economizar custos de armazenamento. Quando você desconhece os padrões de acesso dos objetos ou os padrões de acesso mudam com o passar do tempo, é possível fazer a transição para os objetos para a classe de armazenamento S3 Intelligent-Tiering para economizar automaticamente.

O Amazon S3 da suporte a um modelo de cachoeira para fazer a transição entre classes de armazenamento, conforme mostrado no diagrama a seguir.

É possível adicionar ações de transição a uma configuração do Ciclo de Vida do S3 para solicitar que o Amazon S3 exclua objetos no final de sua vida útil. Quando um objeto chega ao fim de sua vida útil com base em sua configuração de ciclo de vida, o Amazon S3 executa uma ação de expiração com base no estado de versionamento do S3 em que o bucket se encontra:

- Bucket sem versão: o Amazon S3 coloca o objeto em fila para remoção e o remove de forma assíncrona, removendo permanentemente o objeto.
- Bucket com versionamento ativado: se a versão atual do objeto não for um marcador de exclusão, o Amazon S3 adicionará um marcador de exclusão com um ID de versão exclusivo. Isso torna a versão atual desatualizada, e o marcador de exclusão se torna a versão atual.

- Bucket com versionamento suspenso: o Amazon S3 cria um marcador de exclusão com null como ID da versão. Esse marcador de exclusão substitui uma versão do objeto por uma ID de versão nula na hierarquia de versões, o que efetivamente exclui o objeto.
- Quando você adiciona uma configuração do ciclo de vida a um bucket, as regras de configuração se aplicam aos objetos existentes e aos objetos que serão adicionados no futuro. Por exemplo, se você adicionar uma regra de configuração do ciclo de vida hoje com uma ação de expiração que faz com que objetos com um prefixo específico expirem 30 dias após a criação, o Amazon S3 entrará na fila para remover objetos existentes com mais de 30 dias e que tenham o prefixo especificado.

A vida útil (TTL) para DynamoDB é um método econômico para excluir itens que não são mais relevantes. A TTL permite definir um carimbo de data e hora de validade por item que indica quando um item não é mais necessário. O DynamoDB exclui automaticamente os itens expirados alguns dias após o vencimento, sem consumir o throughput de gravação.

- Para usar a TTL, primeiro habilite-a em uma tabela e, depois, defina um atributo específico para armazenar o carimbo de data e hora de vencimento da TTL. O carimbo de data e hora deve ser armazenado no formato de hora de época do Unix na granularidade de segundos. Sempre que um item é criado ou atualizado, é possível calcular o prazo de validade e salvá-lo no atributo TTL.
- Itens com atributos TTL válidos e expirados podem ser excluídos pelo sistema, normalmente alguns dias após a expiração. Você ainda pode atualizar os itens expirados que estão pendentes de exclusão, incluindo alterar ou remover os atributos TTL. Ao atualizar um item expirado, recomendamos usar uma expressão de condição para garantir que o item não tenha sido excluído posteriormente. Use expressões de filtro para remover itens expirados dos resultados de Scan e Query.
- Os itens excluídos funcionam de forma semelhante aos excluídos por meio de operações de exclusão típicas. Depois de excluídos, os itens entram no DynamoDB Streams como exclusões de serviços, em vez de exclusões de usuários, e são removidos dos índices secundários locais e globais, assim como outras operações de exclusão.

Com ElastiCache o for Redis, você pode controlar a atualização de seus dados em cache usando TTLs ou expirando em chaves em cache. Após o término do tempo definido, a chave é excluída do cache e o acesso ao armazenamento de dados de origem é necessário junto com o acesso aos dados atualizados.

- Dois princípios determinam o apropriado TTLs a ser aplicado e os tipos de padrões de cache a serem implementados. Primeiro, é importante que você entenda a taxa de alteração dos dados subjacentes. Em segundo lugar, é importante avaliar o risco de dados desatualizados serem retornados ao seu aplicativo em vez de seus equivalentes atualizados.
- Com dados dinâmicos que mudam com frequência, talvez você queira aplicar menos dados TTLs que expiram a uma taxa de alteração que corresponda à do banco de dados principal. Isso reduz o risco de retornar dados desatualizados e, ao mesmo tempo, fornece um buffer para descarregar as solicitações do banco de dados.
- Também é importante reconhecer que, mesmo que você esteja armazenando dados em cache apenas por minutos ou segundos, em vez de durações mais longas, TTLs a aplicação adequada às chaves em cache pode resultar em um aumento no desempenho e em uma melhor experiência geral do jogador com seu jogo.

Etapas de implementação

- Use as políticas de ciclo de vida do Amazon S3 para fazer a transição de objetos para níveis infrequentes de acesso ou arquivamento e configurar ações de expiração para excluir objetos desnecessários com base nas regras do ciclo de vida.
- Ative o Time to Live (TTL) nas tabelas do DynamoDB para excluir automaticamente itens expirados sem consumir a taxa de transferência de gravação, definindo o timestamp de expiração no Unix epoch time.
- Defina ElastiCache as chaves apropriadas TTLs com base nas taxas de alteração de dados e na tolerância ao risco de dados desatualizados, facilitando a atualização dos dados em cache e melhorando a experiência do jogador.

Hardware e serviços

GAMESUS02: Como você gerencia o uso de recursos computacionais no back-end de seus jogos?

Os estúdios devem desenvolver uma estratégia de computação que use uma combinação de diferentes tipos de computação, serviços gerenciados e plano de economia para otimizar seu uso.

Você também deve otimizar a forma como os servidores de jogos e os serviços de back-end são agrupados nas instâncias de computação para reduzir o número de recursos desnecessários.

Práticas recomendadas

- GAMESUS02-BP01 Selecione serviços gerenciados para cargas de trabalho computacionais apropriadas
- GAMESUS02-BP02 Dimensione corretamente sua computação e implante o desempenho da GPU somente quando necessário

GAMESUS02-BP01 Selecione serviços gerenciados para cargas de trabalho computacionais apropriadas

Arquitete seus serviços de back-end de jogos para usar serviços gerenciados para cargas de trabalho de tráfego altamente variáveis ou orientadas por eventos. Os serviços gerenciados transferem o gerenciamento da infraestrutura AWS e distribuem o impacto ambiental entre vários usuários devido aos planos de controle multilocatários.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

AWS serviços como AWS Lambda, AWS Fargate (Contêineres) e Amazon Gamelift (orquestração de servidores de jogos) podem executar código, contêineres ou orquestrar seus servidores de jogos sem precisar gerenciar a infraestrutura subjacente. Esses serviços são escalonados automaticamente com base na demanda dos jogadores e você só é cobrado pelos recursos que consome. Como a infraestrutura subjacente é gerenciada em seu nome, você pode se concentrar exclusivamente nos requisitos de jogos e serviços de back-end.

É possível usar o AWS Lambda para executar código sem provisionar ou gerenciar servidores. O Lambda executa seu código em uma infraestrutura computacional de alta disponibilidade e executa a administração dos recursos computacionais, incluindo manutenção do servidor e do sistema operacional, provisionamento de capacidade, escalabilidade automática e registro. Com o Lambda, você precisa fornecer seu código em um dos tempos de execução de linguagem compatíveis com o Lambda. O Lambda é útil para processar eventos de jogos, autenticação de jogadores, processamento de compras no jogo e solicitações de matchmaking. O Lambda é escalado automaticamente com base no número de eventos e pode lidar com picos inesperados no tráfego.

AWS Fargateé um mecanismo de computação sem servidor para contêineres que funciona com o Amazon Elastic Container Service (ECS) e o Amazon Elastic Kubernetes Service (EKS). AWS Fargate facilita o foco na criação de seus aplicativos, aliviando a necessidade de provisionar e gerenciar servidores, permite que você especifique e pague por recursos por aplicativo e melhora a segurança por meio do isolamento de aplicativos por design. O Fargate é ideal para serviços de back-end que lidam com perfis de jogadores, gerenciamento de estado e matchmaking.

GameLiftA Amazon é um serviço gerenciado para implantar, operar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões. Você pode implantar seu primeiro servidor de jogos na nuvem em apenas alguns minutos, economizando até milhares de horas de engenharia no desenvolvimento inicial de software e reduzindo os riscos técnicos que geralmente fazem com que os desenvolvedores eliminem os recursos multijogador de seus projetos.

Etapas de implementação

- Use AWS Lambda para cargas de trabalho orientadas por eventos, como processamento de eventos de jogos, autenticação de jogadores, compras no jogo e solicitações de matchmaking, aproveitando seu escalonamento automático e gerenciamento sem servidor.
- Implemente AWS Fargate com ECS ou EKS para serviços de back-end, como perfis de jogadores, gerenciamento de estado e matchmaking, removendo o gerenciamento do servidor e melhorando o isolamento do aplicativo.
- Use GameLift a Amazon para implantar e escalar servidores de jogos dedicados para jogos multijogador baseados em sessões, reduzindo o tempo de desenvolvimento e a complexidade operacional.

GAMESUS02-BP02 Dimensione corretamente sua computação e implante o desempenho da GPU somente quando necessário

Arquitete seus servidores e back-end de jogos para utilizar com eficiência os recursos de computação. O provisionamento excessivo da computação pode gerar custos desnecessários e minimizar a quantidade de recursos ociosos ou subutilizados. As instâncias de GPU devem ser usadas para apoiar esforços específicos de desenvolvimento, como reconstruções de HLOD no Unreal, ou se seus servidores de jogos precisarem delas por design. Isso reduz consideravelmente o impacto ambiental e os custos de suas cargas de trabalho.

Nível de risco exposto se esta prática recomendada não for estabelecida: Alto

Orientação para implementação

Você deve otimizar seus servidores de jogos e serviços de back-end para usar vários tipos de EC2 instâncias e o menor número de instâncias necessário. Isso aumenta o número de instâncias disponíveis para atender às suas necessidades durante o desenvolvimento ou para o lançamento de seus jogos. Você também deve combinar o tipo de instância com a carga de trabalho específica que você está implantando. As instâncias otimizadas para computação oferecem suporte a uma ampla variedade de casos de uso, incluindo servidores de jogos e serviços de back-end, como matchmaking. As instâncias otimizadas para memória são projetadas para oferecer desempenho rápido para cargas de trabalho que processam grandes conjuntos de dados na memória. Use instâncias de GPU conforme necessário para requisitos de alto desempenho, mas não para tarefas gerais de computação. Se possível, arquitete seus serviços ou servidores de jogos para serem executados em ARM com instâncias AWS Graviton. O Graviton é o tipo de instância com melhor desempenho e eficiência energética disponível em. AWS Eles também oferecem desempenho e custos aprimorados quando comparados aos tipos de instância x86.

Use o AWS Compute Optimizerpara identificar as configurações de AWS recursos ideais, como tipos de instância do Amazon Elastic Compute Cloud (EC2), configurações de volume do Amazon Elastic Block Store (EBS), tamanhos de tarefas dos AWS Fargate serviços do Amazon Elastic Container Service (ECS) ativados, AWS Lambda licenças de software comercial, tamanhos de memória funcional e classes de instância de banco de dados Amazon Relational Database Service (RDS), usando aprendizado de máquina para analisar métricas históricas de utilização. O Compute Optimizer fornece um conjunto e uma experiência APIs de console para reduzir custos e aumentar o desempenho da carga de trabalho, recomendando os recursos ideais para suas cargas de trabalho. AWS AWS

Etapas de implementação

- Combine recursos de computação com cargas de trabalho específicas usando instâncias otimizadas para computação para servidores de jogos, instâncias otimizadas para memória para grandes conjuntos de dados e instâncias de GPU somente para tarefas como reconstruções de HLOD ou servidores de jogos dependentes de GPU.
- Otimize a utilização da computação implantando instâncias AWS Graviton sempre que possível para eficiência energética, melhor desempenho e economia de custos em comparação com instâncias x86.
- Use AWS Compute Optimizer para analisar o histórico de utilização e recomendar as configurações mais eficientes para EC2 cargas de trabalho do AWS ECS AWS Lambda e do Amazon RDS para reduzir custos e melhorar o desempenho.

Recursos

Consulte os recursos a seguir para saber mais sobre nossas melhores práticas relacionadas à sustentabilidade.

- ONU: Indústria de jogos destaca ameaças ao planeta
- Meio: Sustentabilidade ambiental no desenvolvimento de jogos: jogando com responsabilidade por um futuro mais verde

Principais AWS serviços

- Amazon Aurora
- Amazon DynamoDB
- Amazon DocumentDB (compatível com MongoDB)
- Amazon ElastiCache
- Amazon S3
- Classes de armazenamento do Amazon S3
- Classe de armazenamento hierárquico inteligente Amazon S3
- Classe de armazenamento Amazon S3 Express One Zone
- Amazon Elastic Block Store
- AWS Lambda
- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service
- Amazon GameLift
- AWS Compute Optimizer

Conclusão

Os jogos são projetados para oferecer experiências de entretenimento a um público global de jogadores e têm características de uso que geralmente são imprevisíveis e variáveis. O Games Industry Lens descreve os tipos comuns de cenários que normalmente compõem uma arquitetura de jogos e fornece um conjunto de perguntas e melhores práticas a serem consideradas ao criar e operar jogos na nuvem. Ao aplicar essa estrutura à sua arquitetura de jogos, você pode criar jogos confiáveis, seguros, eficientes e econômicos na nuvem.

Colaboradores

As pessoas a seguir contribuíram na elaboração deste documento:

- Adam Hatfield, arquiteto sênior de soluções, Amazon Web Services
- Brady Webb, gerente técnico de contas, Amazon Web Services
- Bruce Ross — Arquiteto de soluções sênior, líder da Well-Architected Lens, Amazon Web Services
- Caleb Cecil, arquiteto de soluções associado, Amazon Web Services
- Carlos Perez, Sucesso em Otimização de Nuvem SA, Amazon Web Services
- Chase Herrington, gerente técnico de contas da ESL, Amazon Web Services.
- Chris Blackwell, arquiteto de soluções sênior, Amazon Web Services
- Corey Ouderkirk, gerente técnico sênior de contas, Amazon Web Services
- Derek Villavicencio, Prototipagem SA, Amazon Web Services
- Erik Ynigo Becerril, arquiteto sênior de soluções, Amazon Web Services
- Grzegorz Ochmanski, arquiteto sênior de soluções, Amazon Web Services
- Hadrian Baron, gerente técnico sênior de contas, Amazon Web Services
- Ian Armbruster, gerente sênior de soluções ao cliente, Amazon Web Services
- Jed Obray, gerente técnico sênior de contas, Amazon Web Services
- Khurram Khokhar, gerente técnico sênior de contas, Amazon Web Services
- Kyle Somers, gerente sênior de arquitetura de soluções, Amazon Web Services
- Madhuri Srinivasan, redatora técnica sênior, Well-Architected, Amazon Web Services
- Matthew Wygant, orientador sênior de TPM, Well-Architected, Amazon Web Services
- Nataliya Godunok, Sucesso em Otimização de Nuvem SA, Amazon Web Services
- Nirav Doshi, arquiteto de soluções principal, Amazon Web Services
- Olivia Liddell, arquiteta de soluções, Amazon Web Services
- Randy James, gerente técnico principal de contas, Amazon Web Services
- Reou Ando, arquiteto de soluções de jogos, Amazon Web Services
- Richard Raseley, gerente técnico sênior de contas, Amazon Web Services
- Sam Patzer, arquiteto sênior de soluções, Amazon Web Services
- Scott Selinger, arquiteto de soluções sênior, Amazon Web Services
- Sean Allen, arquiteto de soluções sênior, Amazon Web Services

- Serge Poueme, arquiteto sênior de soluções, Amazon Web Services
- Stewart Matzek, redator técnico sênior, Well-Architected, Amazon Web Services
- Trenton Potgieter, arquiteto AI/ML/Analytics sênior de soluções, Amazon Web Services

Revisões do documento

Para ser notificado sobre atualizações desse whitepaper, inscreva-se no feed RSS.

Alteração	Descrição	Data
Nova versão de lente	Lente inteira atualizada com novas orientações de melhores práticas.	9 de dezembro de 2025
Publicação inicial	Whitepaper publicado pela primeira vez.	19 de novembro de 2021

AWS Glossário

Para obter a AWS terminologia mais recente, consulte o AWS glossário na Glossário da AWS Referência.

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.