

Lente de la industria de juegos



Lente de la industria de juegos: AWS Documento técnico

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Resumen e introducción	i
Disponibilidad de lentes	2
Principios de diseño	3
Escenarios	5
Alojamiento de juegos para una jugabilidad sincrónica en tiempo real	5
Procesos del servidor de juegos	6
Alojamiento de servidores de juegos basado en sesiones con backend sin servidor	8
Arquitectura híbrida y multirregional para juegos de baja latencia	10
Backends de juegos	12
Arquitectura de backend de juegos basada en contenedores	12
Arquitectura de backend de juegos basada en servidores sin servidor	15
Desarrollo de juegos en la nube (CGD)	18
Desarrollo de juegos en la nube: CI/CD	19
Desarrollo de juegos en la nube: estaciones de trabajo	21
Game analytics pipeline (Canalización de análisis de juegos)	23
Definiciones	26
Sistema de juego	27
Servidor de juegos	28
Cliente de juegos	31
Mensajería	31
Operaciones de juego en vivo (Live Ops)	32
Excelencia operativa	34
Principios de diseño	34
Operaciones en vivo	35
GAMEOPS01-BP01 Usa los objetivos del juego y las métricas de rendimiento empresarial para desarrollar tu estrategia de operaciones en vivo	36
Estructura de la cuenta	37
GAMEOPS02-BP01 Adopte una estrategia multicuenta para aislar diferentes juegos y aplicaciones en sus propias cuentas	37
GAMEOPS02-BP02 Organice los recursos de infraestructura mediante el etiquetado de recursos	42
Implementaciones de juegos	42
GAMEOPS03-BP01 Valida y prueba tu infraestructura y sistemas de juego principales existentes antes de reutilizarlos en tu juego	43

GAMEOPS03-BP02 Realiza tareas de ingeniería de rendimiento antes de cada lanzamiento (o al menos en el caso de las versiones principales)	44
GAMEOPS03-BP03 Realice una prueba de carga temprana y frecuente	45
GAMEOPS03-BP04 Adopta una estrategia de despliegue que minimice el impacto en los jugadores	48
GAMEOPS03-BP05 Se necesita una infraestructura preescalable para soportar los picos de demanda	52
Monitorización de la salud	54
GAMESOPS04-BP01 Instrumenta el juego para detectar y monitorear los problemas que afectan a los jugadores	55
Prueba de carga	56
GAMEOPS05-BP01 Elija el escenario, la arquitectura y el marco de pruebas de carga adecuados para cumplir sus objetivos	56
Optimización a lo largo del tiempo	60
GAMEOPS06-BP01 Supervisa las métricas clave del juego para identificar las tendencias y patrones de los jugadores y utiliza la información para mejorar el juego	61
GAMEOPS06-BP02 Actualiza y adapta el enfoque de las pruebas de carga a medida que cambia el juego	62
Recursos	64
Documentación y blogs	64
Soluciones de socios	65
Documentos técnicos	65
Videos	66
Material de formación	66
Seguridad	67
Principios de diseño	68
Aspectos básicos de seguridad	68
GAMESEC01-BP01 Utilice los roles y el acceso federado, en lugar del usuario raíz de la cuenta, para realizar acciones en su entorno AWS	69
GAMESEC01-BP02 Se utiliza para configurar rápidamente un entorno de cuentas múltiples en AWS Control Tower AWS	70
GAMESEC01-BP03 Utilice políticas de roles con privilegios mínimos que se adapten a funciones laborales específicas	72
GAMESEC01-BP04 Utilice las funciones y las políticas de acceso federado junto con las políticas de acceso a nivel de cuenta para conceder el acceso a sus recursos AWS	73
GAMESEC01-BP05 Usa un proveedor de identidad central	74

Seguridad continua	75
GAMESEC02-BP01 Utilice plantillas listas para su implementación para las prácticas de seguridad estándar	76
GAMESEC02-BP02 Utilice técnicas de reparación automatizadas cuando surja un incidente de seguridad	77
Identity and Access Management	78
GAMESEC03-BP01 Determina tu enfoque para identificar y controlar el acceso de los jugadores al entorno y los recursos del juego	79
GAMESEC03-BP02 Autentica las solicitudes que se envían a tu servicio de backend de juegos	81
GAMESEC03-BP03 Usa tu servicio de backend de juegos para validar las solicitudes de los jugadores para unirse a una partida multijugador	83
GAMESEC03-BP04 Aplique una política de seguridad estricta a las cuentas de usuario de los jugadores exigiendo una contraseña segura	84
GAMESEC03-BP05 Ofrece a los jugadores la opción de configurar la autenticación multifactorial (MFA) en sus cuentas	85
Control de acceso	86
GAMESEC04-BP01 Restringe el acceso al contenido descargable a clientes y usuarios autorizados	86
GAMESEC04-BP02 Limita el acceso de origen a las redes de entrega de contenido autorizadas () CDNs	89
GAMESEC04-BP03 Implemente restricciones geográficas para limitar el acceso no autorizado	90
GAMESEC04-BP04 Restrinja el acceso al contenido con soluciones de gestión de derechos digitales (DRM)	91
Detección	92
GAMESEC05-BP01 Implementa una estrategia integral de recopilación de datos para monitorear el comportamiento de los jugadores	93
GAMESEC05-BP02 Recopila, almacena y analiza los registros de uso de los jugadores para detectar comportamientos inapropiados	94
Protección de la infraestructura	95
GAMESEC06-BP01 Utilice herramientas para detectar las amenazas a su infraestructura y responder a ellas	95
GAMESEC06-BP02 Utilice herramientas de inteligencia artificial y aprendizaje automático para automatizar aspectos de su estrategia de protección de infraestructuras	97

GAMESEC06-BP03 Utilice la información de los registros a nivel del sistema para mejorar continuamente su estrategia de protección de la infraestructura	98
Respuesta a incidentes	100
GAMESEC07-BP01 Implemente un plan de respuesta a incidentes para hacer frente a los malos actores y los comportamientos abusivos	100
GAMESEC07-BP02 Prohíbe las cuentas que estén asociadas a personas malintencionadas	101
Seguridad de las aplicaciones	102
GAMESEC08-BP01 Aplique la seguridad en todas las etapas del proceso de CI/CD	103
Automatice la seguridad	104
GAMESEC09-BP01 Integre las herramientas y la automatización para reducir el tiempo medio de las revisiones de seguridad	104
Modelado de amenazas	105
GAMESEC10-BP01 Determine cuándo y cómo realizar los ejercicios de modelado de amenazas a lo largo del ciclo de vida de desarrollo de las aplicaciones	106
Recursos	107
Fiabilidad	109
Principios de diseño	109
Principios básicos	110
Arquitectura de la carga de trabajo	110
GAMEREL01-BP01 Distribuye la infraestructura de juegos en varias zonas y regiones de disponibilidad para mejorar la resiliencia	110
Administración de cambios	113
GAMEREL02-BP01 Implementa una estrategia de escalado que incorpore el estado de las sesiones de juego de los jugadores activos	113
GAMEREL02-BP02 Support el uso de varios tipos de EC2 instancias para tu juego	114
Administración de errores	115
GAMEREL03-BP01 Supervise las interrupciones en los servidores de juegos y utilice los datos para mejorar la arquitectura de alojamiento a fin de alcanzar los objetivos de confiabilidad	116
GAMEREL03-BP02 Implementa una combinación flexible de funciones del juego para gestionar los fallos con un impacto mínimo en la experiencia del jugador	118
GAMEREL03-BP03 Supervisa los eventos de la infraestructura a lo largo del tiempo para medir el impacto en el comportamiento de los jugadores	120
Recursos	122
Eficiencia del rendimiento	123

Principios de diseño	123
Selección de la arquitectura	124
GAMEPERF01-BP01 Evalúe los requisitos de recursos y las necesidades de escalabilidad del servidor de juegos	125
GAMEPERF01-BP02 Tenga en cuenta la sobrecarga operativa a la hora de escalar los servidores de juegos	126
GAMEPERF01-BP03 Evalúe la integración con otros AWS servicios, entornos de desarrollo, arquitecturas de CPU objetivo y funciones	128
Selección de región	129
GAMEPERF02-BP01 Selecciona una región local que esté cerca de tus jugadores	130
GAMEPERF02-BP02 Diseñe un enfoque que permita colocar la infraestructura de juegos sensible a la latencia cerca de los jugadores para mejorar el rendimiento	131
Desarrollo iterativo	134
GAMEPERF03-BP01 Usa GameLift Amazon Anywhere y un kit de herramientas de prueba GameLift	134
GAMEPERF03-BP02 Pruebe el rendimiento y la escalabilidad de los servidores de juegos	136
GAMEPERF03-BP03 Optimice la utilización de los recursos de los contenedores GameLift	137
Computación y hardware	138
GAMEPERF04-BP01 Supervisa los procesos del servidor de juegos para detectar problemas	139
GAMEPERF04-BP02 Pon a prueba el rendimiento de tu servidor de juegos con escenarios de juego simulados y reales	140
Calcular la selección	141
GAMEPERF05-BP01 Compara el rendimiento de tus juegos en varios tipos de cómputo	141
GAMEPERF05-BP02 Transfiere las tareas non-latency-sensitive informáticas a flujos de trabajo asíncronos	143
Administración de datos	145
GAMEPERF06-BP01 Centraliza la recopilación y el almacenamiento de registros	145
GAMEPERF06-BP02 Clasifica y almacena los datos del juego en función de los patrones de acceso	146
GAMEPERF06-BP03 Permite formatear y agrupar registros de forma eficiente	147
GAMEPERF06-BP04 Implemente políticas de rotación y retención de registros	147
GAMEPERF06-BP05 Utiliza herramientas de monitorización y visualización	148
Redes y entrega de contenido	148

GAMEPERF07-BP01 Define los umbrales de latencia de red para tu juego	149
GAMEPERF07-BP02 Ejecuta un servicio de emparejamiento independiente para cada modo de juego y región de alojamiento del juego	149
GAMEPERF07-BP03 Supervisa periódicamente el rendimiento del emparejamiento	150
GAMEPERF07-BP04 Supervisa periódicamente el rendimiento de la red	151
GAMEPERF07-BP05 Utilice la tecnología de aceleración de red para mejorar el rendimiento en Internet	152
Proceso y cultura	154
GAMEPERF08-BP01 Informa e incluye al jugador en tu proceso	154
GAMEPERF08-BP02 Alinee la selección de soluciones con las habilidades y la experiencia del equipo de ingeniería	156
Recursos	156
Optimización de costos	159
Principios de diseño	160
Práctica de administración financiera en la nube	161
Conocimiento del gasto y del uso	161
GAMECOST01-BP01 Implementa la atribución del coste por jugador, función del juego y entorno	161
GAMECOST01-BP02 Descubra oportunidades de optimización	163
Recursos rentables	164
GAMECOST02-BP01 Optimice el costo de la transferencia de datos a través de Internet	165
GAMECOST02-BP02 Optimiza el número de sesiones de juego alojadas en cada instancia del servidor de juegos para optimizar los costes	167
GAMECOST02-BP03 Seleccione la opción de precio de computación adecuada para reducir los costos	168
Costos de transferencia de datos	171
GAMECOST03-BP01 Elija el tipo de almacenamiento apropiado para el contenido generado por el usuario a fin de reducir los costos	172
GAMECOST03-BP02 Optimiza las bases de datos para los backends de los juegos	173
Gestión de los recursos de oferta y demanda	175
Optimización a lo largo del tiempo	175
Recursos	175
Sostenibilidad	177
Principios de diseño	177
Selección de región	178
Alineación con la demanda	178

Software y arquitectura	178
Administración de datos	178
GAMESUS01-BP01 Usa tecnologías de almacenamiento que se adapten a los patrones adaptados al contenido del usuario, la información de los suscriptores y las compras dentro del juego	179
GAMESUS01-BP02 Usa las políticas de ciclo de vida o de caducidad del TTL para eliminar los juegos, los datos de usuario, los archivos de registro o los activos obsoletos que no sean necesarios	181
Hardware y servicios	184
GAMESUS02-BP01 Seleccione los servicios gestionados para las cargas de trabajo informáticas adecuadas	184
GAMESUS02-BP02 Ajusta el tamaño de tu ordenador y despliega el rendimiento de la GPU solo donde sea necesario	186
Recursos	187
AWS Servicios clave	187
Conclusión	189
Colaboradores	190
Revisiones del documento	192
AWS Glosario	193
.....	cxciv

Lente de la industria de los juegos: AWS Well-Architected Framework

Fecha de publicación: 9 de diciembre de 2025 ([Revisiones del documento](#))

El [AWS Well-Architected](#) Framework ayuda a los arquitectos de la nube a crear una infraestructura segura, de alto rendimiento, resiliente y eficiente para sus aplicaciones y cargas de trabajo. Basado en seis pilares (excelencia operativa, seguridad, confiabilidad, eficiencia del rendimiento, optimización de costos y sustentabilidad), Well-Architected ofrece un enfoque coherente para que los clientes AWS y socios evalúen las arquitecturas, remedien los riesgos e implementen diseños que brinden valor empresarial.

Desde este punto de vista, nos centramos en cómo diseñar, implementar y diseñar las cargas de trabajo de sus juegos en el. Nube de AWS Definimos los componentes, exploramos los escenarios de carga de trabajo más comunes y describimos los principios de diseño que le ayudarán a aplicar el marco de Well-Architected. Le recomendamos que comience a diseñar su arquitectura teniendo en cuenta las prácticas recomendadas y las preguntas del documento técnico [AWS Well-Architected Framework](#). Este documento proporciona prácticas recomendadas complementarias para los clientes del sector de los videojuegos.

Este enfoque especifica las mejores prácticas destinadas a abordar las características únicas de la creación y el funcionamiento de juegos en la nube, sobre la base de nuestra experiencia trabajando con desarrolladores y editores del sector de los videojuegos de todo el mundo. Proporciona orientación sobre cómo diseñar y operar su entorno de forma que se optimice los costes y se adapte a las fluctuaciones de la demanda mundial de jugadores. Este objetivo también proporciona orientación para proteger la infraestructura del juego y ajustar el rendimiento a fin de ofrecer una experiencia de juego positiva.

Este documento está dirigido a quienes desempeñan funciones tecnológicas, como los directores de tecnología (CTOs), los directores técnicos de los estudios de videojuegos, los arquitectos, los desarrolladores y los miembros del equipo de operaciones. Tras leer este documento, comprenderás las AWS mejores prácticas y estrategias que debes utilizar al diseñar arquitecturas para juegos.

Disponibilidad de lentes

Los lentes personalizados amplían la guía de mejores prácticas proporcionada por AWS Well-Architected Tool. AWS WA Tool le permite crear sus propias [lentes personalizadas](#) o utilizar lentes creadas por otras personas que hayan compartido con usted.

Para empezar a revisar la carga de trabajo de tus juegos, descarga e importa [Games Industry Lens](#) AWS Well-Architected Tool desde el repositorio público de lentes personalizadas de [AWS Well-Architected](#). GitHub

Principios de diseño

El AWS Well-Architected Framework identifica los siguientes principios generales de diseño para facilitar un buen diseño en la nube para las cargas de trabajo de juegos:

- Comprenda el comportamiento y los patrones de uso de los jugadores para hacer que el juego evolucione y ayude a protegerlos: para impulsar la mejora continua de su juego y gestionar de forma eficaz la experiencia del jugador, es importante tener visibilidad de cómo interactúan los jugadores con el juego en sí y con el resto de los jugadores. Esto te ayudará a entender cómo mejorar el juego, gestionar los costes y controlar y reaccionar ante cualquier actividad de uso no autorizado que suponga un riesgo para la experiencia del jugador.
- Utiliza tecnologías que simplifiquen las operaciones del juego y aceleren el desarrollo: da prioridad a la adopción de tecnologías que puedan mejorar la velocidad y reducir la sobrecarga operativa que supone ofrecer nuevas funciones y mejoras a los jugadores. Los juegos se basan en los éxitos y hay muchas opciones que los jugadores deben tener en cuenta, por lo que actuar con rapidez y adaptarse a los cambios es fundamental para el éxito de un juego. Considere si se siente cómodo manejando su propio software o si prefiere adoptar un servicio gestionado comparable de parte de AWS AWS nuestros socios o de ambos.
- Optimice su arquitectura para mejorar las métricas que reflejen la experiencia real de los jugadores: a medida que vaya adaptando y evolucionando su arquitectura con el paso del tiempo, piense en cómo afectarán estas mejoras y cambios a la experiencia de los jugadores. Las cargas de trabajo de los juegos deberían poder resistir y minimizar el impacto de los fallos para bloquear las interrupciones generalizadas del juego. Las funciones y los sistemas del juego que no dependan de manera crítica unos de otros deberían estar desacoplados para reducir el impacto de los fallos y aislar los problemas que afectan a los jugadores.
- Diseñe la infraestructura para cumplir con los picos de concurrencia de jugadores y escalarla dinámicamente según sea necesario: la infraestructura debe diseñarse para adaptarse a la demanda de los jugadores. Las métricas, como la simultaneidad de las sesiones de los jugadores y el número de inicios de sesión, se pueden utilizar para escalar de forma preventiva antes de que los sistemas se sobrecarguen. Las métricas de uso reactivo del sistema, como el consumo de CPU y memoria, se pueden utilizar para escalar después de que los sistemas se sobrecarguen. Al escalar tu infraestructura de forma dinámica, puedes reducir los costes de funcionamiento del juego.
- Implemente manuales de ejecución para mejorar las operaciones del juego: los manuales operativos deben utilizarse para gestionar de forma coherente las tareas operativas del juego

que se repiten. Deberían existir guías para los flujos de trabajo habituales de las operaciones del juego, como investigar y responder a las denuncias de los jugadores, gestionar las actividades de preescalado de la infraestructura a fin de prepararse para eventos a gran escala previstos, como el lanzamiento de nuevas temporadas y el lanzamiento de contenido de los juegos, y para abordar las actividades habituales de mantenimiento de los juegos.

Escenarios

En esta sección, cubrimos varios escenarios que son comunes en la arquitectura de un juego. Cada escenario incluye las características comunes que impulsan el diseño y un ejemplo de diagrama de arquitectura de referencia.

Escenarios

- [Alojamiento de juegos para una jugabilidad sincrónica en tiempo real](#)
- [Backends de juegos](#)
- [Arquitectura de backend de juegos basada en servidores sin servidor](#)
- [Desarrollo de juegos en la nube \(CGD\)](#)
- [Game analytics pipeline \(Canalización de análisis de juegos\)](#)

Alojamiento de juegos para una jugabilidad sincrónica en tiempo real

El juego sincrónico en tiempo real permite que dos o más jugadores participen e interactúen en un juego simultáneamente, donde el estado del juego se comparte entre los jugadores conectados para crear una experiencia lo más parecida posible a la de tiempo real. Algunos ejemplos de juegos sincrónicos son los juegos de disparos en primera persona, los juegos multijugador masivos en línea (MMOG), los juegos de deportes y acción o los juegos en línea en los que dos o más jugadores deben estar conectados para compartir la experiencia de juego prácticamente en tiempo real.

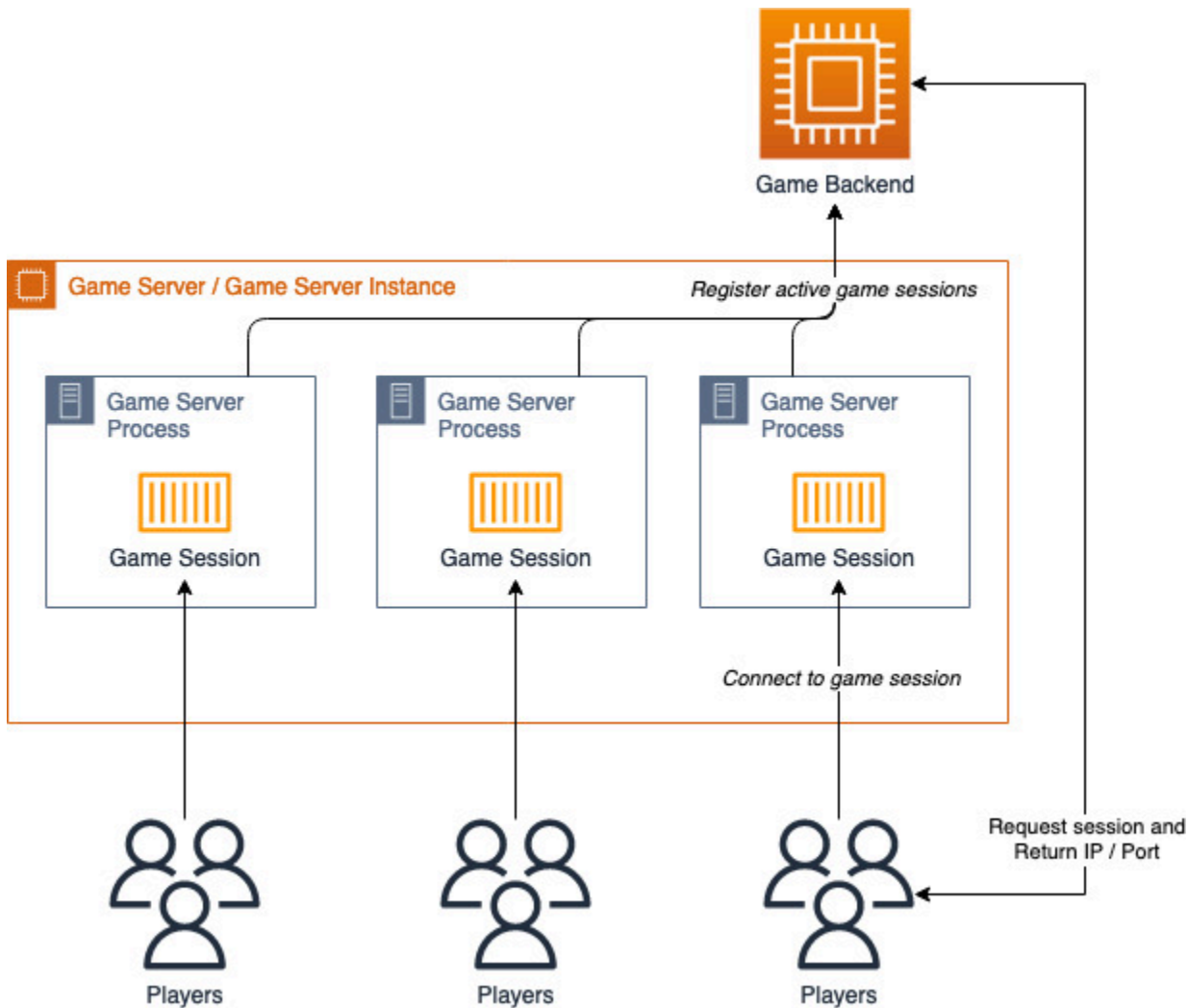
Las características de las arquitecturas de juego sincrónico en tiempo real incluyen:

- Algunos juegos se pueden alojar como sesiones de juego mediante procesos del servidor de juegos que se ejecutan en servidores dedicados. Algunos juegos pueden usar arquitecturas tipo P2P que emplean utilidades de recorrido de sesión más livianas para NAT (STUN) o de transmisión transversal mediante relés alrededor de servidores NAT (TURN). Independientemente del tipo de servidor de que se trate, los servidores de juegos se alojan en varios centros de datos y en todo el mundo. Regiones de AWS
- Los clientes del juego pueden unirse a una sesión de juego solicitando una partida a un servicio centralizado de búsqueda de parejas alojado en el sistema de administración del juego o seleccionando una partida de una lista predefinida de servidores de juegos disponibles. El cliente del juego cuenta con una dirección IP y un puerto al que conectarse.

- Muchos juegos sincrónicos son sensibles a la latencia, como los juegos de disparos en primera persona y los juegos multijugador masivos en línea. Es probable que incluyan algoritmos como el rebobinado y la dilatación del tiempo para minimizar los efectos de retardo, pero también pueden tener una tolerancia a la latencia predefinida que se mide y optimiza cuidadosamente para reducir el retardo que a veces pueden experimentar los jugadores en situaciones de alta latencia. Esta información de latencia se determina al hacer que los clientes del juego hagan ping al servidor de juego disponible en las Regiones de AWS para capturar métricas como la latencia, la fluctuación de la red y otras métricas importantes para la experiencia de juego. Estas métricas se envían a un servicio central de recopilación de métricas en el sistema central del juego para que las transmisiones de operaciones en directo puedan monitorizar el estado del juego. Durante el proceso de emparejamiento, los clientes del juego proporcionan sus datos de latencia actuales como uno de los parámetros de solicitud al solicitar una partida, y el servicio de emparejamiento puede utilizar esos datos de latencia como una de las variables a la hora de seleccionar un servidor de juego para alojar al jugador.
- Por lo general, el juego se desarrolla a través de una combinación de protocolos (por ejemplo, los servidores de juegos utilizan una mensajería basada en UDP más rápida, con emparejamiento, autenticación y otro tipo de tráfico cliente-servidor mediante HTTPS).
- Los servidores de juegos utilizan algoritmos y diseños para minimizar el tráfico cliente-servidor, como la transmisión de transformaciones, los deltas y la compresión de datos.
- Los servidores de juegos suelen ser blanco de actividades maliciosas y deberían protegerse con una solución de protección DDo tipo S. AWS Shield Advanced

Procesos del servidor de juegos

El siguiente diagrama ilustra una arquitectura de servidor de juegos típica. Describe la relación lógica entre una instancia de servidor de juegos y los procesos del servidor de juegos que alojan las sesiones de juego.



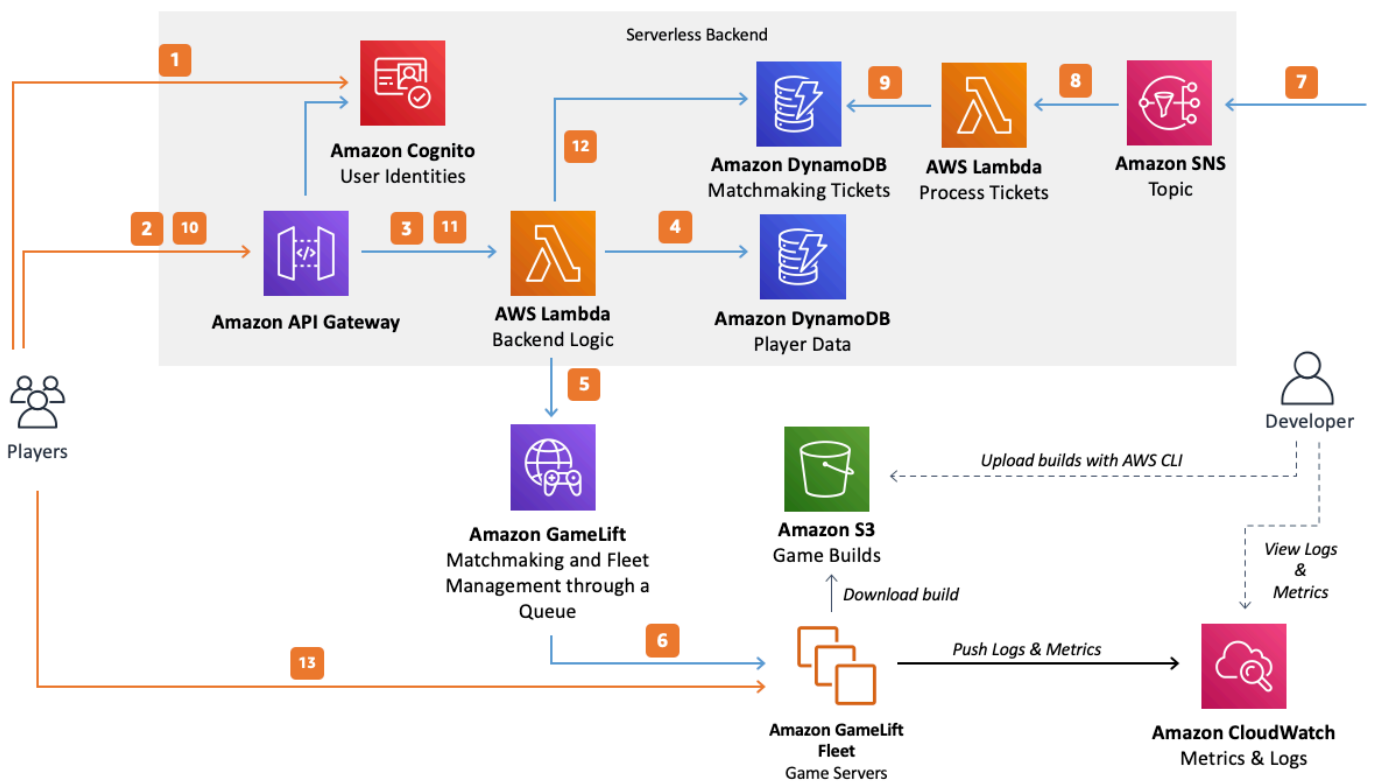
Arquitectura de servidor de juegos lógico

- Una EC2 instancia de Amazon se utiliza como servidor de juegos, también conocida como instancia de servidor de juegos. El servidor de juegos aloja uno o más procesos del servidor de juegos y cada uno ejecuta una copia de la versión del servidor de juegos. Por lo general, se ejecutan varios procesos del servidor de juegos en una instancia de servidor de juegos para utilizar los recursos informáticos de forma eficiente y reducir los costes. Cuando una sesión de juego está activa y lista para albergar sesiones de jugadores, su estado se actualiza con el backend del juego (normalmente un servicio de emparejamiento) para que pueda empezar a usarse como anfitrión de jugadores.
- El backend del juego puede proporcionar al jugador la dirección IP y el puerto del servidor que aloja una sesión de juego para que pueda conectarse para jugar.

Alojamiento de servidores de juegos basado en sesiones con backend sin servidor

Al desarrollar una arquitectura para tu juego, ten en cuenta las funciones y capacidades que necesitas y el nivel de sobrecarga de administración operativa que estás dispuesto a soportar. Para ofrecer el mejor equilibrio entre facilidad de operación y flexibilidad, puedes desarrollar tu juego con los servicios gestionados de los proveedores de servicios en la nube. Los servicios gestionados te permiten desarrollar y personalizar tus propias funciones de juego personalizadas y, al mismo tiempo, reducen la carga que supone implementar y gestionar la infraestructura.

Para organizar una partida multijugador basada en sesiones, es necesario disponer de una infraestructura de servidor que aloje los procesos del servidor del juego, así como de un backend escalable para gestionar las sesiones y las partidas. La siguiente arquitectura de referencia muestra cómo se pueden utilizar el alojamiento GameLift gestionado por Amazon y un backend sin servidor para gestionar tus juegos basados en sesiones.



Alojamiento GameLift gestionado por Amazon para juegos basados en sesiones

El diagrama describe el proceso de convertir a los jugadores en juegos que se ejecutan en un alojamiento de juegos GameLift gestionado. Incluye los siguientes pasos:

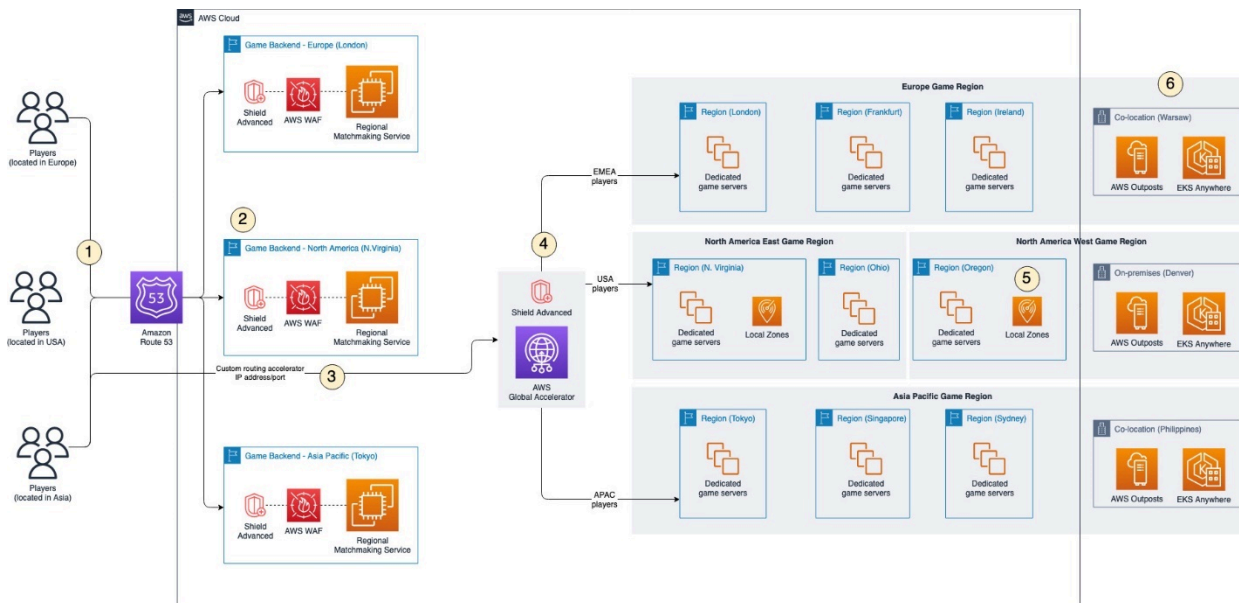
1. El cliente del juego solicita una identidad de Amazon Cognito de un grupo de identidades de Amazon Cognito. De forma opcional, se puede conectar a proveedores de identidad externos.
2. El cliente del juego recibe credenciales de acceso temporales y solicita una sesión de juego a través de una Amazon API Gateway firmando la solicitud con las credenciales de Amazon Cognito.
3. API Gateway invoca una AWS Lambda función.
4. La función Lambda solicita datos de los jugadores de una tabla de Amazon DynamoDB. La identidad de Amazon Cognito se utiliza para solicitar de forma segura los datos correctos del jugador, ya que la identidad autenticada se proporciona en los datos de contexto de la solicitud.
5. Al utilizar los datos correctos del jugador para obtener información adicional (como el nivel de habilidad del jugador), la función Lambda solicita una partida mediante GameLift FlexMatch el matchmaking. Puede definir una configuración de FlexMatch emparejamiento con documentos de configuración basados en JSON. El cliente del juego puede generar métricas de latencia haciendo ping a los puntos finales de los servidores en varias regiones, y los datos de latencia se pueden utilizar para respaldar el emparejamiento basado en la latencia.
6. Cuando FlexMatch empareja a un grupo adecuado de jugadores con una latencia adecuada con una región, solicita que la sesión de juego se sitúe en una cola. GameLift La cola contiene flotas con una o más ubicaciones regionales registradas.
7. Cuando la sesión se coloca en una de las ubicaciones de la flota, se envía una notificación de evento a un tema de Amazon SNS.
8. Una función Lambda recibirá el evento Amazon SNS y lo procesará.
9. Si el mensaje de Amazon SNS es un MatchmakingSucceeded evento, la función Lambda escribe el resultado en DynamoDB con el puerto del servidor y la dirección IP. Se utiliza un valor time-to-live (TTL) para garantizar que los tickets de matchmaking se eliminen de DynamoDB cuando ya no se necesiten.
- 10 El cliente del juego envía una solicitud firmada a API Gateway para comprobar el estado del ticket de emparejamiento en un intervalo específico.
- 11 API Gateway invoca una función Lambda que comprueba el estado del ticket de emparejamiento.
- 12 La función Lambda comprueba DynamoDB para determinar si el ticket se ha realizado correctamente. Si se ha realizado correctamente, la función Lambda devuelve la dirección IP, el puerto y el ID de sesión del reproductor al cliente. Si el ticket ha fallado, la función Lambda envía una respuesta declarando que la partida no está lista.

13 El cliente del juego se conecta al servidor del juego mediante TCP o UDP mediante el puerto y la dirección IP proporcionados por el servidor. Envía el identificador de sesión del jugador al servidor del juego y el servidor del juego lo valida mediante el SDK de Amazon GameLift Server.

Como alternativa, puede modificar la arquitectura anterior para usar API Gateway WebSockets con Amazon GameLift. En este enfoque, la comunicación entre el cliente del juego y el servicio de backend del juego se produce mediante una [implementación WebSocket basada en](#) el software. Esta implementación se puede utilizar para que la función Lambda del backend del juego inicie un mensaje del lado del servidor al cliente del juego a través de un modelo de sondeo, en lugar de implementar WebSocket un modelo de sondeo.

Arquitectura híbrida y multirregional para juegos de baja latencia

En esta sección se describe una arquitectura híbrida y multirregional para juegos de baja latencia.



Reducir la latencia con la aceleración de la red y la implementación de servidores de juegos en todo el mundo

1. Los jugadores de un juego disponible en todo el mundo pueden proceder de cualquier parte. Cuando un jugador solicita una sesión de juego o un partido, su cliente de juego envía una solicitud al servicio de backend del juego registrado en Amazon Route 53. El enrutamiento basado en la latencia de Route 53 se puede usar para dirigir al jugador al servidor de juego disponible más cercano.

2. El backend del juego se despliega en las poblaciones más cercanas a los jugadores. Regiones de AWS Cada servidor de juego incluye un servicio de búsqueda de partidas regional que busca una sesión de juego en todas las regiones del juego. Aunque la solicitud de emparejamiento de un jugador es procesada por un servicio de emparejamiento regional cercano al jugador, el servicio de emparejamiento es capaz de redirigir a un jugador a una sesión de juego en otra región del juego, si es necesario. Esta acción mejora la resiliencia y el rendimiento. Además, cada servicio de back-end del juego utiliza AWS WAF y AWS Shield Advanced proporciona filtros web de nivel 7 y control de bots, además de protección contra los ataques de denegación de servicio (DDoS) por distribución. Tienes muchas opciones para crear tu servicio de backend de juegos, por ejemplo, sin servidor, contenedores, EC2 instancias o alojar el servicio de backend del juego en tus propios centros de datos.
3. Para mejorar la experiencia de los jugadores reduciendo la latencia y la inestabilidad de la red, se ha implementado un acelerador de enrutamiento personalizado con AWS Global Accelerator, que optimiza automáticamente el enrutamiento del tráfico desde el cliente del juego al servidor del juego a través de la red global. AWS El [acelerador de enrutamiento personalizado](#) se configura para asignar los puertos de escucha de Global Accelerator al puerto de instancia del servidor de juegos. EC2 El cliente del juego se conecta a la IP y al puerto de Global Accelerator que sirve de proxy y dirige a los jugadores de forma determinista a la IP y el puerto correctos del servidor de juego que alojan la sesión de juego.
4. Tu juego incluye regiones de juegos lógicos fáciles de usar para los jugadores que representan un conjunto de ubicaciones de servidores de juegos que están geográficamente cerca unas de otras, como América del Norte o Asia-Pacífico. Para reducir la latencia y aumentar la cobertura geográfica, puedes utilizar una combinación de diferentes soluciones de alojamiento de servidores de juegos para mejorar la experiencia del jugador. Prioriza su uso siempre que Regiones de AWS sea posible, ya que estas ubicaciones cuentan con todas las funciones y son las que ocupan una mayor capacidad.
5. Usa las Zonas AWS Locales para alojar servidores de juegos en ubicaciones geográficas de jugadores desatendidos en las que no dispongas de instalaciones de alojamiento existentes o en las que no Región de AWS haya ninguna disponible.
6. AWS Outposts Implíquelos en sus centros de datos locales y proveedores de ubicación compartida existentes para crear un plano de control y una experiencia de administración sin fisuras en cada ubicación de despliegue mediante racks y servidores montados en rack totalmente gestionados. AWS Outposts también son útiles si no dispone de capacidad de servidor existente en su entorno local. Sin embargo, si desea una implementación híbrida con software que se ejecute en su propia infraestructura de servidores existente, debe usar Amazon EKS Anywhere, que le permite crear y ejecutar clústeres de Kubernetes en su propia infraestructura con conectividad de vuelta a

Amazon EKS. Esto proporciona una vista de consola coherente de sus clústeres de Kubernetes en las instalaciones.

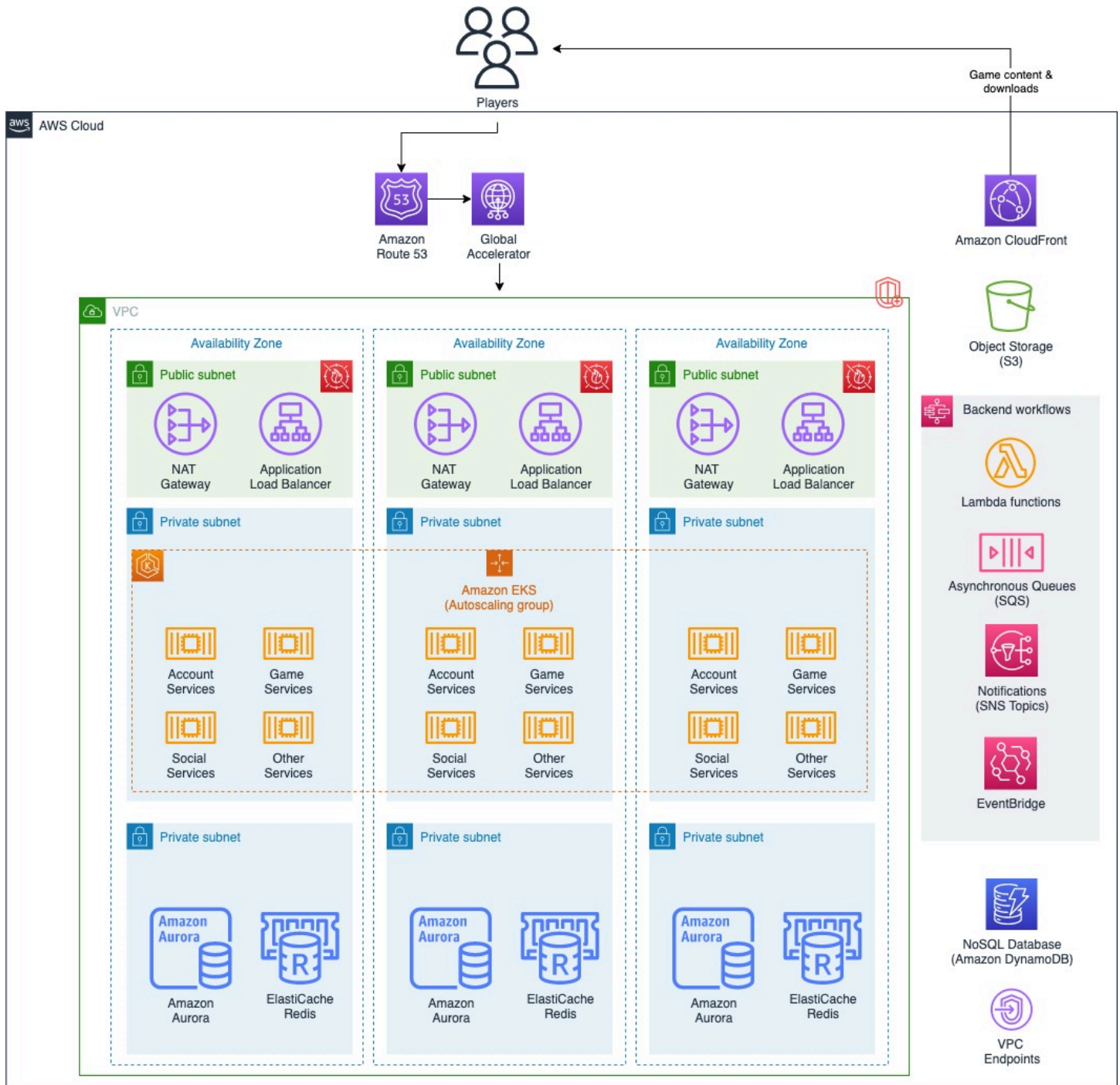
Backends de juegos

Los backends de los juegos se utilizan para administrar el estado del juego y del jugador, así como para integrar funciones sociales y de sistema en el juego que respaldan la experiencia de juego. La gestión del perfil de los jugadores, el almacenamiento de objetos e inventario y las estadísticas y tablas de clasificación son ejemplos de servicios alojados en los backends de los juegos.

Los backends de los juegos suelen crearse como REST, a los APIs que los clientes acceden mediante HTTPS. Sin embargo, también son habituales otros enfoques, como los WebSockets que proporcionan canales bidireccionales para casos de uso, como las notificaciones a los clientes sobre el chat y la presencia en el juego. Los backends de los juegos se pueden implementar mediante una variedad de arquitecturas de implementación diferentes, que incluyen instancias, contenedores o una arquitectura sin servidor.

Arquitectura de backend de juegos basada en contenedores

En esta sección se describe una arquitectura de backend de juegos basada en contenedores.



Hospedar un backend de juego mediante contenedores

- Los jugadores acceden al juego mediante un software cliente del juego, que se les puede distribuir a través de un sistema de juego, una tienda digital o una descarga directa desde una red de entrega de contenido (CDN), como Amazon. CloudFront CDNs proporcionan almacenamiento en caché en ubicaciones periféricas para acelerar el rendimiento de los usuarios que descargan

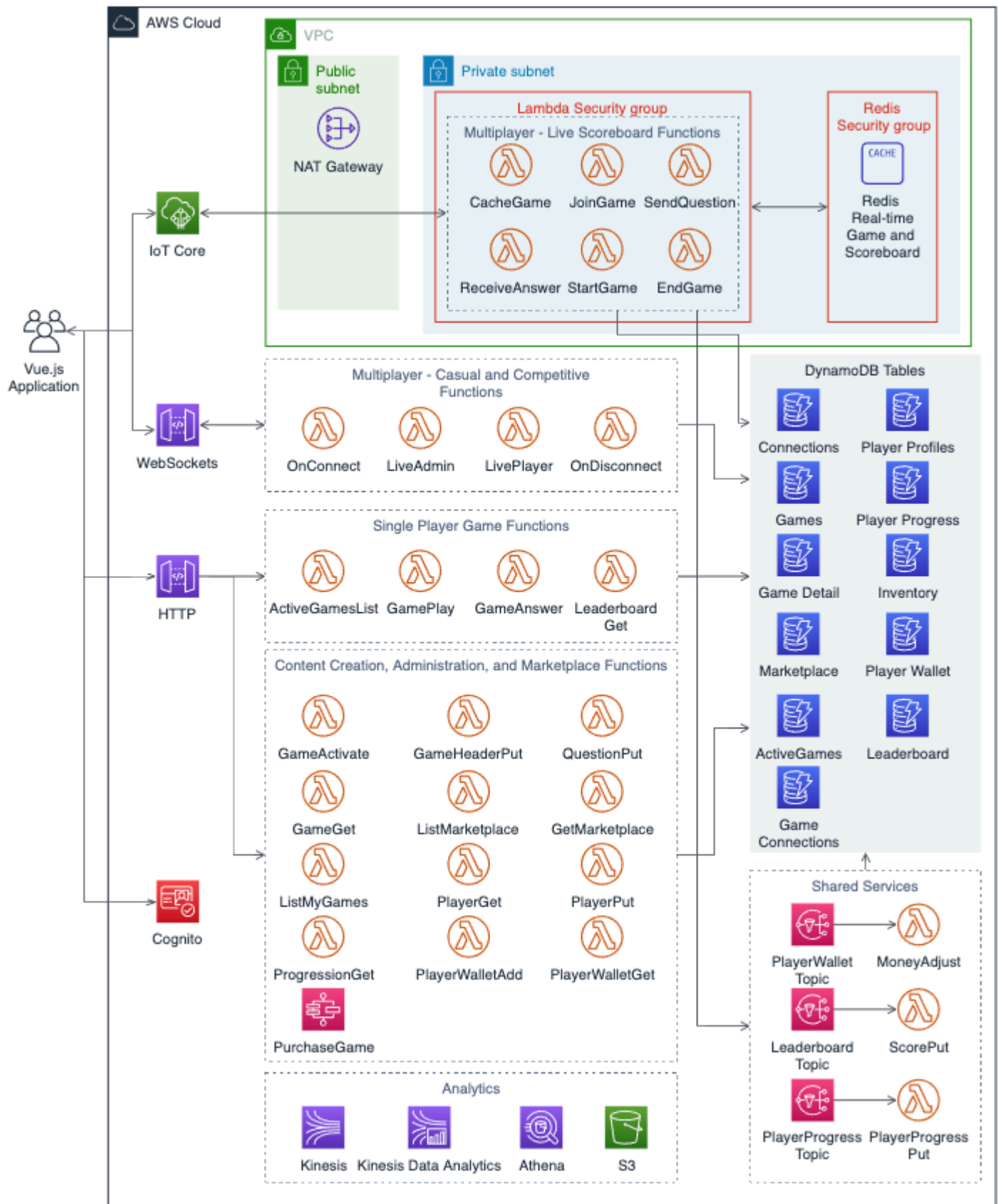
contenido. Por ejemplo, se CloudFront puede usar para distribuir el software del cliente del juego a sus jugadores, así como los activos del juego y otro contenido.

- AWS Global Accelerator proporciona una aceleración del tráfico y controles personalizables para enrutar el tráfico de los clientes de los juegos a los balanceadores de carga, así como para enrutar el tráfico entre regiones con fines multirregionales y de conmutación por error. Los nombres de dominio personalizados para el REST del backend del juego APIs están configurados en Amazon Route 53 para dirigir el tráfico a los puntos de enlace de Global Accelerator. Si no se muestra en el diagrama, AWS Shield Advanced puede proporcionar una mitigación de DDoS adicional para el acelerador y el backend del juego.
- NAT Gateway y Application Load Balancer se implementan en las subredes públicas de cada una de las zonas de disponibilidad utilizadas por el backend del juego para ofrecer una alta disponibilidad en la región. AWS WAF se implementa en el Application Load Balancer para proporcionar un filtrado de tráfico web de capa 7.
- El backend del juego se aloja como un conjunto de microservicios individuales basados en contenedores implementados en un clúster de Amazon EKS en subredes privadas repartidas por zonas de disponibilidad para mayor resiliencia. El escalado automático ajusta de forma dinámica la capacidad de los servicios y los nodos del clúster en función del uso de los recursos, lo que normalmente se correlaciona con la demanda de los jugadores. Mientras que el escalador automático de clústeres ajusta automáticamente la cantidad de nodos del clúster, el escalador automático de módulos horizontales escala automáticamente los módulos desplegados en el clúster.
- Los datos del juego y de los jugadores se almacenan en bases de datos y cachés de fondo que se despliegan en subredes privadas de todas las zonas de disponibilidad y se replican entre los nodos principales y de réplica. Amazon Aurora es una opción popular para casos de uso como perfiles de jugadores, derechos y compras dentro del juego, que pueden tener requisitos de consulta más complejos y pueden beneficiarse de las capacidades de modelado de datos relacionales de MySQL y PostgreSQL. Amazon ElastiCache es útil para crear tablas de clasificación y pub/sub mensajería de alto rendimiento y para almacenar en caché los datos a los que se accede con frecuencia a fin de reducir la latencia y la carga en las bases de datos. Amazon DynamoDB es un almacén de datos NoSQL totalmente gestionado que resulta ideal para patrones de acceso impredecibles y la capacidad de escalarse hasta alcanzar un rendimiento prácticamente ilimitado para casos de uso como datos de estado de jugadores y juegos, datos de sesión, almacenes de inventario y artículos, o casos de uso en los que desee una base de datos global con una sobrecarga mínima.

- Los flujos de trabajo de procesamiento asíncrono se deben utilizar para realizar trabajos que puedan realizarse en segundo plano, como actualizar tablas de clasificación o enviar solicitudes de amistad. Configure el backend de su juego para incluir este tipo de trabajo en las colas de Amazon SQS para escalarlo a medida que su juego crezca o considere la posibilidad de utilizar temas de Amazon SNS para distribuir el trabajo entre muchas colas de aplicaciones de consumo para su procesamiento en paralelo. Utilice las funciones AWS Lambda para realizar el procesamiento de una manera basada en eventos para reducir los costos de infraestructura informática y los gastos generales de administración. En el caso de los flujos de trabajo de larga duración o que requieran la coordinación de tareas en varios pasos, considere la posibilidad de organizar todo el flujo de trabajo utilizando AWS Step Functions. Amazon EventBridge puede utilizarse para iniciar funciones que respondan a eventos de AWS servicio y aplicaciones personalizadas.

Arquitectura de backend de juegos basada en servidores sin servidor

Muchos desarrolladores de juegos no quieren gestionar la infraestructura, sino que prefieren crear sus juegos con tecnologías que les permitan centrarse en el software. En este escenario, se recomienda una arquitectura sin servidor porque permite crear y lanzar funciones con mayor rapidez y con menos sobrecarga operativa. Las arquitecturas sin servidor se diseñan con servicios en la nube que pueden escalarse dinámicamente en función de la demanda sin necesidad de configurar, administrar ni escalar los servidores. La siguiente arquitectura de referencia ilustra cómo crear un juego con una arquitectura sin servidor.



Arquitectura de referencia del backend de juegos basada en servidores sin servidor

Esta arquitectura de referencia ilustra un juego de preguntas y respuestas basado en la web que ofrece funciones para un jugador y multijugador.

- **Autenticación del jugador:** los jugadores se autentican mediante Amazon Cognito, que proporciona una autenticación segura con un directorio de usuarios para la gestión de la identidad de los jugadores.
- **La lógica del juego como funciones sin servidor:** las funciones del juego y la lógica empresarial interna funcionan como AWS Lambda funciones que se inician en respuesta a eventos, lo que reduce los costes, ya que solo se paga cuando se ejecuta la función. Lambda le ofrece la flexibilidad de escribir cada función del juego como un microservicio independiente mediante el lenguaje de programación que prefiera. Por ejemplo, puede optar por desarrollar funciones Lambda de .NET si tiene experiencia en el uso de C# para crear juegos de Unity, o puede optar por desarrollar funciones Lambda de Node.js si desea programar un frontend y un backend para un juego basado en la web tanto en JavaScript
- **Almacén de datos NoSQL para datos de juegos y jugadores:** utilice DynamoDB para almacenar sus datos de jugadores y juegos, ya que está diseñado específicamente para almacenar grandes cantidades de datos de microservicios. Como se ilustra en esta arquitectura, se recomienda utilizar almacenes de datos independientes para las necesidades de almacenamiento de datos de cada función del juego, lo que facilita la supervisión y la administración de las funciones de forma independiente. Esto también crea límites de separación si el propietario de una función o servicio cambia dentro de tu equipo. En esta arquitectura de referencia, las tablas de DynamoDB se utilizan para almacenar datos como el estado de la conexión, los detalles del juego, el progreso de los jugadores y la información de la tabla de clasificación.
- **Jugabilidad para un jugador:** las funciones para un jugador permiten a los jugadores realizar acciones como seleccionar y jugar una partida y ver la tabla de clasificación. Estas funciones se implementan como servicios de RESTful backend alojados en una API HTTP de Amazon API Gateway que invoca la función Lambda adecuada para obtener y configurar datos en tablas de DynamoDB. Cuando finaliza el juego, el backend también envía notificaciones a los temas de Amazon SNS que inician las funciones de Lambda de forma asíncrona para almacenar el progreso y las estadísticas del jugador.
- **Jugabilidad multijugador:** las características del juego multijugador requieren que los jugadores puedan interactuar con el juego para point-to-point comunicarse, así como transmitir y recibir actualizaciones de otros jugadores conectados. Una WebSockets implementación es adecuada para la point-to-point comunicación en un juego ligero, como un juego de preguntas y respuestas.

Los jugadores pueden establecer una WebSockets conexión con Amazon API Gateway WebSockets, que administra la conexión y solo invoca las funciones Lambda cuando hay mensajes que enviar o recibir para un jugador. Para los casos de uso en los que es necesaria la one-to-many comunicación entre jugadores, AWS IoT Core ofrece soporte para la mensajería mediante WebSockets MQTT, lo que permite a los clientes suscribirse a los temas y actuar en función de los mensajes que reciben. En esta arquitectura, se utilizan WebSockets más MQTT para admitir casos de uso, como la transmisión en directo de actualizaciones del juego y la formulación de preguntas a los jugadores conectados. Como alternativa AWS IoT, puedes elegir Redis Pub/Sub para la entrega de mensajes o Redis Streams si necesitas retener los mensajes.

- Utilice funciones de Lambda habilitadas para VPC para acceder a los recursos de sus subredes privadas: configure las funciones de Lambda habilitadas para VPC para acceder a los recursos de las subredes privadas de su VPC, como ElastiCache Amazon, que se utiliza para reducir los tiempos de consulta de conjuntos de datos de baja latencia, como las tablas de clasificación en vivo.

[Para obtener más información, consulte la Guía para el alojamiento personalizado del backend de juegos en. AWS](#)

Desarrollo de juegos en la nube (CGD)

El desarrollo de juegos en la nube (CGD) se refiere a la infraestructura y las herramientas necesarias para el ciclo de vida del desarrollo de un juego para crear, probar y desarrollar un juego. El desarrollo de juegos es colaborativo entre los usuarios y los requisitos de infraestructura cambian con frecuencia a lo largo de las etapas de desarrollo.

Muchos desarrolladores de juegos están optando por equipos de desarrollo remotos y distribuidos por todo el mundo, lo que requiere una tecnología que respalde este tipo de desarrollo. Los desarrolladores de juegos pueden alojar todos o parte de estos entornos AWS y utilizar la disponibilidad global de ellos para acercar los recursos Regiones de AWS a los usuarios y gestionar sus entornos de desarrollo de forma más rentable escalando la computación y el almacenamiento según sea necesario.

Los entornos pueden variar en función de las necesidades de los desarrolladores de juegos, pero suelen incluir estaciones de trabajo para desarrolladores en las que artistas, diseñadores, ingenieros, responsables del control de calidad, contratistas y demás personal pueden realizar su trabajo. Estos entornos también suelen incluir una granja de compilaciones compuesta por repositorios de

código fuente para que los usuarios comprueben sus cambios y la CI/CD infraestructura para crear, empaquetar y probar los artefactos desarrollados.

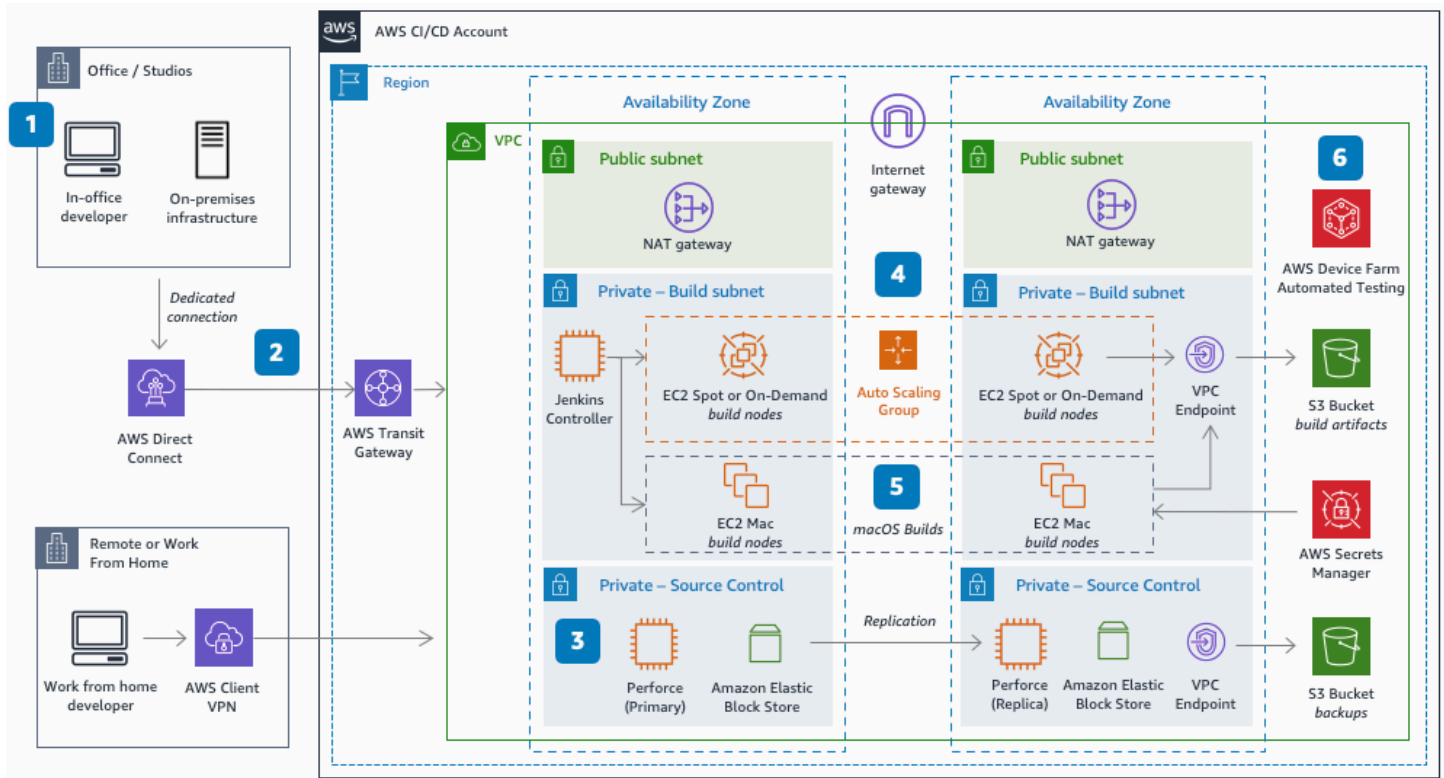
Estas arquitecturas de producción de juegos tienen las siguientes características:

- Los usuarios deben poder acceder a una estación de trabajo virtual a través de un navegador web o un cliente de escritorio local, como [Amazon DCV](#), que les proporciona una sesión de streaming de baja latencia para acceder al mismo software y herramientas a los que tendrían acceso si estuvieran trabajando en una máquina de una oficina o estudio de desarrollo. Estas estaciones de trabajo virtuales, normalmente un servidor basado en la nube, deberían permitir al usuario colaborar y trabajar en sus proyectos completamente en un entorno de nube a través de una LAN o WAN. Cuando los usuarios no utilizan las máquinas de forma activa, se debe hacer una copia de seguridad de su trabajo en un almacenamiento duradero en la nube, por ejemplo, un repositorio de control de código fuente o un [sistema de archivos como Amazon Elastic File System \(EFS\)](#) y [Amazon FSx](#), y sus máquinas deben apagarse para reducir los costes.
- [Los repositorios de control de código fuente, como Perforce, deben diseñarse con alta disponibilidad mediante la replicación entre zonas de disponibilidad o entre entornos locales con copias de seguridad almacenadas en almacenamiento en la nube, como Amazon S3.](#) Por ejemplo, un servidor Perforce basado en la nube debe incluir un servidor de confirmación principal alojado en una zona de disponibilidad con replicación en un servidor en espera alojado en otra zona de disponibilidad de la misma región.
- Los recursos de la granja de compilaciones para el desarrollo de juegos deben diseñarse con un escalado automático para que los recursos de cómputo se aprovisionen cuando sean necesarios, y [las instancias EC2 puntuales](#) deben usarse para reducir los costos incurridos al ampliar la cantidad de servidores necesarios para las compilaciones.

Desarrollo de juegos en la nube: CI/CD

CI/CD La infraestructura es importante a la hora de desarrollar juegos, independientemente del tamaño del equipo, para mejorar los tiempos de iteración, la fiabilidad de los equipos, un despliegue eficiente y un mejor control del proceso de desarrollo y lanzamiento a fin de ofrecer a los jugadores una experiencia de juego de alta calidad. Un proceso de desarrollo CI/CD de juegos suele estar compuesto por servidores de control de código fuente y almacenamiento de alta disponibilidad, recursos informáticos para ejecutar las compilaciones y software para realizar pruebas automatizadas, además de la conectividad de red adecuada desde las máquinas de desarrollo. La siguiente arquitectura de referencia muestra cómo descargar las compilaciones de juegos de

entornos de desarrollo de juegos remotos o locales para ayudar a los desarrolladores Nube de AWS a migrar o crear nuevas granjas de compilaciones.



Descarga las compilaciones del juego en la nube

1. AWS Direct Connect proporciona una conexión privada, dedicada y de baja latencia AWS para desarrolladores internos. Los desarrolladores remotos utilizan tecnologías de confianza cero Acceso verificado de AWS, como, por ejemplo, redes privadas virtuales (VPN), como AWS Client VPN.
2. AWS Transit Gateway simplifica la administración de la red para la conectividad entre VPCs y desde las redes locales.
3. Perforce gestiona el control de fuentes y versiones (CI) con el respaldo del almacenamiento de Amazon EBS para obtener datos persistentes y de acceso rápido. Perforce Helix Core está disponible en AWS Marketplace
4. Las confirmaciones inician una compilación (CD) en Jenkins cuando los desarrolladores insertan cambios en Perforce vinculados a una rama. Perforce inicia el envío de una carga útil de JSON a Jenkins. El controlador Jenkins invoca los comandos CLI de Engine Headless para ejecutar y paralelizar el proceso de compilación en nodos Docker efímeros (como Amazon Spot EC2 Instances o Amazon On-Demand Instances). EC2 Los desarrolladores pueden aumentar la disponibilidad mediante el uso de dos controladores Jenkins, uno en cada zona de disponibilidad,

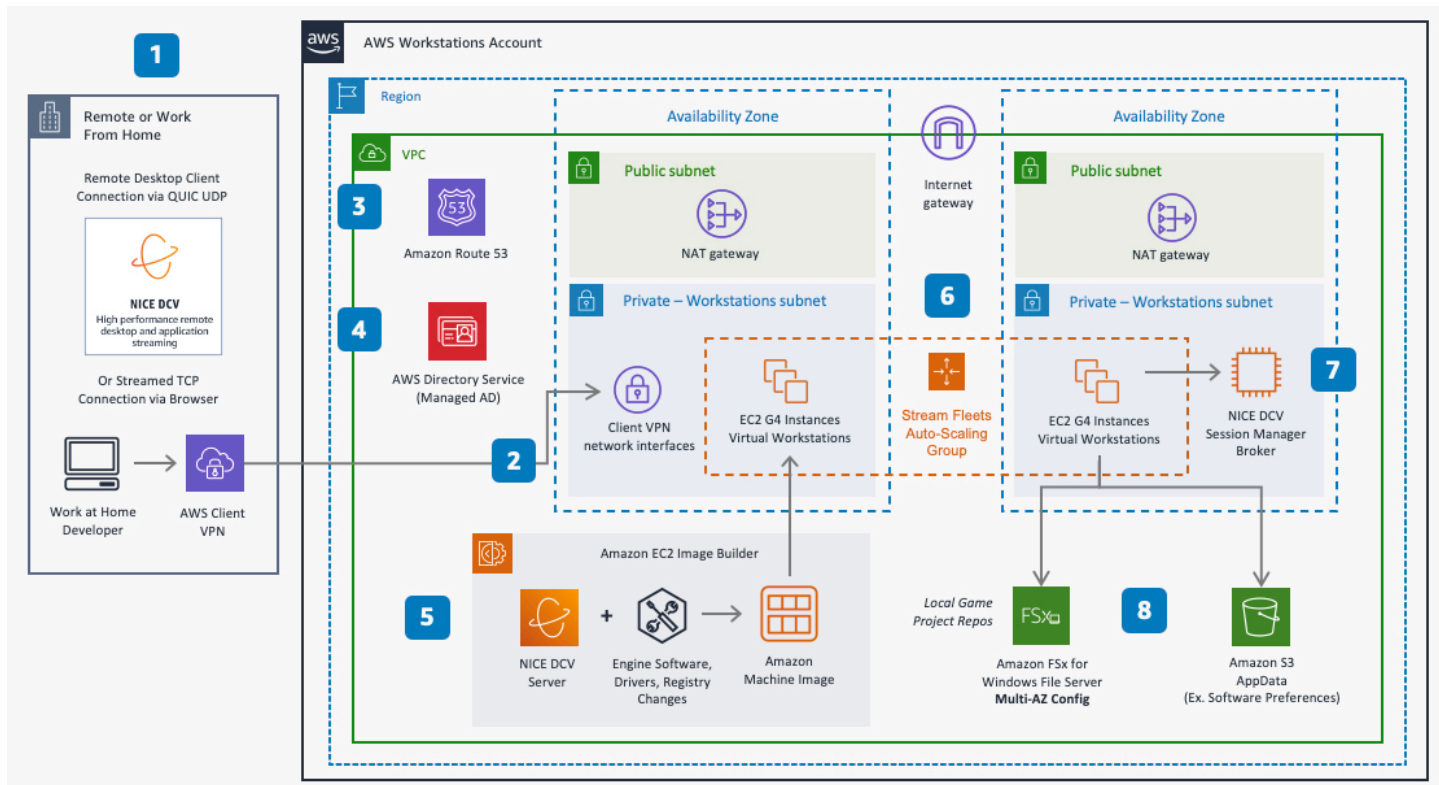
detrás de un balanceador de carga. En el caso de algunos motores de juegos, es posible que los desarrolladores necesiten una infraestructura de licencias adicional configurada en subredes adicionales para vender licencias para el contexto de compilación cada vez que se ejecute una compilación simultánea.

5. La parte de Xcode de las compilaciones de iOS se descarga a las instancias de Amazon EC2 Mac para firmar, compilar y exportar el archivo IPA, lo que divide el proceso y reduce los tiempos de compilación. AWS Secrets Manager contiene perfiles de aprovisionamiento, claves privadas y certificados.
6. Los artefactos de compilación se envían a Amazon S3, que envía notificaciones de éxito o error. AWS Device Farm permite realizar pruebas automatizadas para dispositivos móviles.

Desarrollo de juegos en la nube: estaciones de trabajo

El desarrollo de juegos suele implicar que equipos distribuidos trabajen de forma remota desde varios lugares, lo que requiere acceso a una infraestructura compartida y la capacidad de respaldar un desarrollo colaborativo y disperso geográficamente. Esta tendencia hacia un proceso de desarrollo de juegos más distribuido, que incluye el uso de work-for-hire estudios y el trabajo remoto, exige la implementación de tecnologías y flujos de trabajo sólidos que faciliten la productividad, el intercambio de conocimientos y las prácticas de desarrollo ágiles.

La siguiente arquitectura de referencia demuestra cómo utilizarla AWS para alojar estaciones de trabajo de desarrollo de juegos remotas mediante el protocolo Amazon DCV.



Transmita el desarrollo de juegos desde cualquier lugar con Amazon DCV

1. Amazon DCV es un protocolo de transmisión que admite la transmisión de 4K a 60 FPS. Los desarrolladores que utilizan un navegador se conectan mediante conexiones TCP, mientras que los clientes de escritorio pueden utilizar QUIC UDP a través del puerto 8443 para aumentar el rendimiento.
2. Los desarrolladores utilizan AWS Client VPN para una conexión segura a las interfaces de red de las subredes de las estaciones de trabajo con traducción de direcciones de red de origen (SNAT).
3. Amazon Route 53 proporciona un DNS privado para los recursos de la VPC, así como el reenvío de DNS entrante y saliente.
4. Directory Service proporciona Microsoft Active Directory administrado para permitir el almacenamiento local de proyectos de juegos asignado a usuarios individuales.
5. Las estaciones de trabajo se crean con una Amazon Machine Image (AMI) creada con Image Builder. Las imágenes incluyen Amazon DCV Server, software para desarrolladores, cambios en el registro y controladores, como los controladores de juegos o los controladores de periféricos de NVIDIA. AWS Marketplace incluye los de AMIs uso común para estaciones de trabajo.
6. Las flotas de estaciones de trabajo utilizan gráficos, tipos de EC2 instancias de Amazon que proporcionan GPUs, y se escalan mediante grupos de EC2 Auto Scaling.

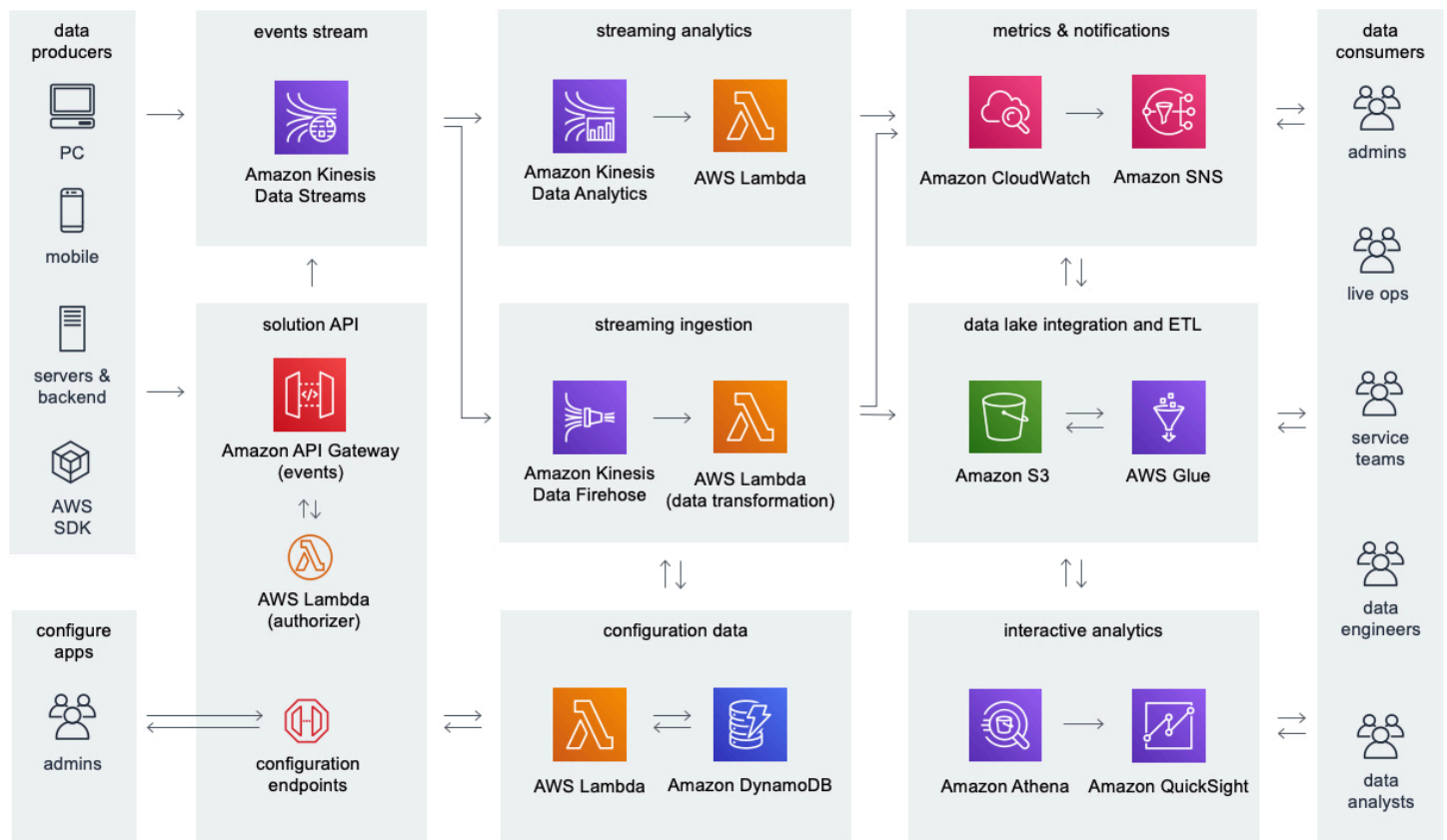
7. El gestor de sesiones de Amazon DCV permite gestionar las sesiones de Amazon DCV.
8. El almacenamiento local de archivos de los proyectos se aloja en Amazon FSx para Windows File Server. Los desarrolladores se comprometen a crear una canalización de CI/CD independiente, pasando del almacenamiento local al control de código fuente.

Game analytics pipeline (Canalización de análisis de juegos)

Los desarrolladores de juegos buscan cada vez más formas de entender mejor el comportamiento de los jugadores para poder mejorar la experiencia de juego y retener y aumentar su base de jugadores. La analítica del juego representa la infraestructura técnica y los procesos necesarios para comprender y analizar los datos que se generan a partir del juego y los servicios relacionados. Por lo general, esto requiere el uso de una arquitectura de canalización de análisis que pueda respaldar este end-to-end proceso, como la implementación de la solución [Game Analytics Pipeline](#).

Las arquitecturas de análisis de juegos tienen las siguientes características:

- Las fuentes de datos envían los datos en un formato común, como JSON, y suelen incluir servidores y servicios de backend de juegos, así como clientes de juegos, como ordenadores, dispositivos móviles y consolas de juegos.
- Un proceso de análisis de juegos automatiza todo el flujo de trabajo consistente en incorporar y almacenar los datos sin procesar y procesarlos en formatos de salida utilizables para que los consumidores de datos, como los usuarios finales y las aplicaciones de análisis, puedan analizarlos de manera eficiente y rentable.
- Los canales de análisis de juegos permiten incorporar y procesar grandes volúmenes de datos en tiempo real para ampliarlos a medida que el juego crece.
- Brinda soporte para casos de uso de informes por lotes y en tiempo real. Por ejemplo, los equipos de operaciones en vivo suelen utilizar paneles y alertas en tiempo real para supervisar la infraestructura del juego y el comportamiento de los jugadores a fin de detectar problemas. Los equipos de analistas de datos suelen utilizar informes por lotes y según sea necesario para comprender las tendencias a lo largo del tiempo.



Canal de análisis de juegos sin servidor para telemetría de juegos

Los datos del juego se obtienen de los clientes, servidores y otras aplicaciones. Los datos de streaming se incorporan a Amazon S3 para la integración del lago de datos y el análisis interactivo. El análisis de streaming procesa los eventos en tiempo real y genera métricas. Los consumidores de datos analizan los datos de métricas en Amazon CloudWatch y los eventos sin procesar en Amazon S3.

- **API de la solución y datos de configuración:** utilice Amazon API Gateway para proporcionar una API REST para administrar su proceso de análisis de juegos y almacenar los datos de configuración en Amazon DynamoDB mediante funciones de Lambda. Puede crear un portal interno sobre esta API o una interfaz de línea de comandos personalizada para la administración. Una API REST también proporciona autenticación de servidor para ingerir datos de juego de fuentes de datos y reenviar los datos de telemetría a Amazon Kinesis Data Streams para procesarlos en tiempo real e incorporarlos al almacenamiento.
- **Transmisión de eventos:** Amazon Kinesis Data Streams captura los datos de streaming de su juego y permite que Amazon Data Firehose y Amazon Managed Service para Apache Flink los procesen en tiempo real.

- **Análisis de streaming:** Managed Service for Apache Flink analiza los datos de eventos de streaming de Kinesis Data Streams y puede generar métricas y alertas personalizadas que se publican CloudWatch mediante las funciones de Lambda.
- **Métricas y notificaciones:** usa Amazon CloudWatch para monitorear las métricas, los registros y las alarmas de tu solución. Utilice Amazon SNS para enviar notificaciones a los ingenieros de guardia y a otros consumidores de datos.
- **Ingestión de streaming:** utilice Firehose para consumir los datos de streaming de Kinesis Data Streams y envíelos a su lago de datos en Amazon S3 para almacenarlos a largo plazo, transformarlos e integrarlos con otros datos.
- **Integración de lagos de datos y ETL:** AWS Glue utilícelo para procesar flujos de trabajo mediante ETL y para organizar sus metadatos en ellos AWS Glue Data Catalog, lo que constituye la base de un lago de datos para su integración con herramientas de análisis flexibles.
- **Análisis interactivos:** los usuarios finales pueden utilizar Amazon Athena para realizar consultas interactivas ad hoc en los conjuntos de datos almacenados en Amazon S3, y Quick Suite se puede utilizar para crear paneles.

Consulte [Game Analytics Pipeline](#) para obtener una implementación de referencia automatizada de una canalización de análisis que se pueda implementar en su cuenta mediante AWS CloudFormation

Definiciones

El AWS Well-Architected Framework se basa en seis pilares: excelencia operativa, seguridad, confiabilidad, eficiencia del rendimiento, optimización de costos y sustentabilidad. AWS proporciona varios componentes principales que te permiten diseñar state-of-the-art arquitecturas para las cargas de trabajo de tus juegos. En esta sección, presentaremos un resumen de las definiciones clave.

A los efectos de este paper, la arquitectura de un juego abarca la infraestructura técnica de backend necesaria para construir y operar un juego. Es posible que algunos juegos no tengan funciones sociales, multijugador u otras funciones en línea y que no requieran el uso de ciertos aspectos de la infraestructura técnica de backend que se describen en este paper. Para ver un análisis detallado de los distintos tipos de cargas de trabajo que se utilizan con frecuencia para dar soporte a la arquitectura de un juego, consulta Escenarios.

La Nube de AWS infraestructura se basa en regiones y zonas de disponibilidad.

- Una región es una ubicación física en el mundo en la que tenemos varias zonas de disponibilidad.
- Las zonas de disponibilidad constan de uno o más centros de datos discretos, cada uno con alimentación, redes y conectividad redundantes, alojados en instalaciones independientes.

En función de las características del juego, es posible que desees desplegar algunos componentes de la arquitectura del juego en varias regiones, por ejemplo, para mejorar el rendimiento de los jugadores o para ofrecerles experiencias personalizadas en función de su ubicación.

Hay muchos tipos diferentes de juegos, y la infraestructura técnica de fondo necesaria para dar soporte a un juego varía según el tipo de juego que se esté desarrollando. Por ejemplo, los tipos de juegos más populares pueden incluir los juegos de disparos en primera persona (FPS), los juegos de rol (RPG), los juegos multijugador masivos en línea (MMOG), los battle royales (BR), los juegos de deportes, los juegos de rompecabezas y más. También hay diferentes modos de interacción que influyen en la arquitectura del juego, como el juego por turnos y el juego simultáneo, con diferentes características de rendimiento.

Los juegos están diseñados para jugarse en uno o más sistemas de juego, incluidos los de escritorio, la web, los dispositivos móviles, las consolas y los modos de interacción más nuevos, como la realidad aumentada (AR), la realidad virtual (VR) y las soluciones de transmisión de juegos. Los juegos suelen ser compatibles con varios sistemas, lo que significa que los jugadores pueden

guardar su progreso en el juego y reanudar el juego en otros sistemas, así como iniciar sesiones de juego con jugadores de otros sistemas.

La monetización de los videojuegos permite a los editores generar ingresos mediante diferentes estrategias, como la publicidad, las compras de juegos digitales y minoristas, las compras de contenido descargable (DLC) dentro del juego, conocidas como microtransacciones, y mediante suscripciones de pago obligatorias para jugar al juego. Algunos de los indicadores clave de rendimiento (KPIs) más comunes en la industria de los videojuegos incluyen:

- Usuarios activos diarios (DAU)
- Usuarios activos mensuales (MAU)
- Usuarios simultáneos (CCU)
- Duración de la sesión
- Coste por instalación (CPI)
- Valor de vida útil del jugador (LTV)
- Ingresos medios por usuario (ARPU)

Sistema de juego

Los videojuegos están desarrollados para jugarse en un sistema de juego que proporciona controles de entrada para el cliente, gráficos, software cliente (conocido como cliente del juego) y hardware y, en algunos casos, funciones exclusivas del sistema para respaldar el juego.

Por lo general, los sistemas de juego se dividen en las siguientes categorías:

- Consolas: sistemas de entretenimiento diseñados específicamente para jugar, incluidos ejemplos populares como Sony, PlayStation Microsoft Xbox y Nintendo Switch. Las consolas ofrecen la posibilidad de jugar mediante la instalación de contenido físico o distribuido digitalmente en el hardware de la consola fabricado por el proveedor del sistema de juego. Según esta definición, una consola puede ser portátil o fija y estar destinada a ser utilizada en un entorno de entretenimiento doméstico.
- Ordenador personal (PC): juegos que se juegan con un software informático instalado en una máquina cliente y que el jugador puede personalizar. Por esta razón, los juegos de PC son populares entre los jugadores debido a la flexibilidad y el control que proporcionan.

- **Web:** juegos que están diseñados para jugarse con un navegador web y que, por lo general, ofrecen la ventaja de permitir al jugador acceder al juego independientemente de su sistema operativo.
- **Móvil:** juegos que se han desarrollado para jugarse en un teléfono móvil, normalmente el sistema operativo de un teléfono inteligente. Por lo general, los juegos móviles se descargan de una tienda de aplicaciones digitales y se instalan en el teléfono.

Además de los sistemas mencionados anteriormente, también hay sistemas incipientes que todavía son relativamente nuevos y están en crecimiento, y tienen una cuota de mercado mucho menor en comparación con los sistemas más predominantes. Algunos ejemplos de sistemas de juego de esta categoría son la realidad aumentada, la realidad virtual y la transmisión de juegos, a veces denominados juegos en la nube.

La transmisión de juegos implica renderizar el juego en la nube y transmitirlo a un cliente ligero, normalmente un navegador. La transmisión de juegos permite a un jugador jugar a un juego que está alojado completamente de forma remota, normalmente en la nube por un proveedor de servicios de transmisión de juegos. En la transmisión de juegos, el jugador se conecta a un juego basado en la nube a través de un navegador o un cliente ligero proporcionado por el proveedor de servicios de juegos en la nube (sistema de juego).

Servidor de juegos

Los servidores de juegos representan uno de los aspectos más importantes de la infraestructura informática del juego. Los servidores de juegos, a veces denominados servidores de juegos dedicados, se utilizan al desarrollar un juego multijugador o cuando se requiere un procesamiento autorizado por parte del servidor de los eventos del juego. El servidor del juego está en el centro de la arquitectura del juego y es el lugar donde se ejecuta la lógica principal, que incluye la gestión del estado del jugador y del juego, así como la gestión de las interacciones entre los clientes de juego conectados y el servidor de juegos. El servidor de juegos suele ser uno de los aspectos más importantes de la arquitectura de un juego, ya que es responsable de procesar las entradas del cliente de juego del jugador y distribuirlas adecuadamente a otros jugadores conectados en tiempo real. Un servidor de juegos con un rendimiento deficiente afecta al rendimiento general de la experiencia de juego. Por lo tanto, debes optimizar el rendimiento del servidor de juegos y proporcionar la capacidad suficiente, especialmente durante el lanzamiento del juego o en los períodos de mayor actividad.

A los efectos de este documento, un servidor de juegos o una instancia de servidor de juegos se refiere a la computación, como una máquina virtual, que aloja uno o más procesos del servidor de juegos. Un proceso de servidor de juegos representa una instancia única de la versión de tu servidor de juegos que aloja una sesión de juego, que es una instancia de tu juego en ejecución a la que los jugadores pueden conectarse a través de una sesión de jugador. Por este motivo, solemos hacer referencia indistintamente al proceso del servidor de juegos o a la sesión de juego debido a la relación implícita de uno a uno entre una sesión de juego y el proceso del servidor de juegos que la aloja. Existen varias opciones informáticas para alojar servidores de juegos, que permiten acceder a una capacidad escalable basada en la nube mediante un aprovisionamiento flexible de los recursos.

AWS

Amazon EC2 proporciona servidores virtuales basados en la nube, conocidos como instancias, compatibles con varias versiones de Linux y Windows. Puede crear instancias y administrarlas directamente como si fuera otro servidor o máquina virtual. Por lo general, se implementan varios procesos de servidores de juegos en una instancia para mejorar la eficiencia y reducir los costos. Amazon EC2 es una buena opción para los servidores de juegos si deseas tener el mayor control sobre la infraestructura informática.

Amazon GameLift ofrece una solución totalmente gestionada para el alojamiento de servidores de juegos dedicados en la nube, así como funciones adicionales como el emparejamiento con GameLift FlexMatch. GameLift proporciona una capa de abstracción superior a la de Amazon EC2 para facilitar la administración de los servidores de juegos y está disponible en la mayoría de los casos Regionales de AWS para que puedas alojar los servidores de juegos cerca de los jugadores y reducir la latencia, lograr una alta disponibilidad y reducir significativamente los costos mediante el uso de instancias puntuales. Si bien se GameLift puede integrar en los backends de juegos existentes, es especialmente útil para los desarrolladores de juegos que no desean desarrollar sus propias soluciones de gestión de servidores de juegos y emparejamiento y prefieren una solución que sea administrada por AWS y que pueda escalar a medida que su juego crece.

Amazon Elastic Container Service (Amazon ECS) es un servicio de organización de contenedores totalmente gestionado que puede utilizar para ejecutar contenedores basados en Docker. También puede usar Amazon Elastic Kubernetes Service (Amazon EKS) para ejecutar contenedores basados en Docker creados con Kubernetes. El uso de tecnologías de contenedores, como las que proporcionan Amazon ECS y Amazon EKS, puede ayudarle a mejorar el uso de la computación al empaquetar de manera eficiente muchos procesos de servidores de juegos u otras instancias de aplicaciones de juegos en una EC2 instancia.

El uso de contenedores también puede mejorar la productividad de los desarrolladores, ya que aloja las aplicaciones con la misma imagen de Docker que utilizan los desarrolladores en sus máquinas locales durante el desarrollo. Puede reducir aún más la sobrecarga operativa mediante AWS Fargate el uso de una solución informática sin servidor para ejecutar contenedores y que es compatible con Amazon EKS y Amazon ECS. Fargate es ideal para los casos de uso en los que quieres ejecutar servidores de juegos en contenedores sin tener la responsabilidad de operar las instancias subyacentes en las que se ejecutan los contenedores.

Puede utilizarlos AWS Outposts para ejecutar la AWS infraestructura y los servicios en un centro de datos o en una instalación local, lo que permite que los juegos se ejecuten en entornos locales y AWS utilicen la misma infraestructura para respaldar una estrategia de adopción de la nube híbrida. AWS Las Zonas Locales funcionan como extensiones Regiones de AWS que permiten que tus servidores de juegos y otras cargas de trabajo sensibles a la latencia se ejecuten más cerca de tus jugadores o equipos de desarrollo. Además, para reducir la latencia de la red global de tus servidores de juegos, puedes usar AWS Global Accelerator para mejorar el rendimiento del tráfico de jugadores a tus servidores de juegos.

AWS Lambda es un servicio informático sin servidor que ejecuta código sin aprovisionar ni administrar los servidores, lo que lo hace útil para casos de uso asíncronos de servidores de juegos, como los juegos por turnos o aquellos que tienen requisitos de procesamiento livianos, una base de código pequeña y donde la funcionalidad del juego se puede diseñar mediante una arquitectura de microservicios sin estado. Es importante tener en cuenta que las funciones de Lambda se ejecutan en función de eventos y por solicitud, en lugar de ejecutar un proceso de servidor de juegos de larga duración. Lambda proporciona la mayor abstracción en tiempo de ejecución de las opciones de este paper, ya que la aplicación subyacente está fácilmente disponible para que los desarrolladores la elijan para alojar su código.

A la hora de elegir el enfoque para el alojamiento de servidores de juegos, tenga en cuenta varios requisitos, como la sobrecarga operativa, las bases de código heredadas, los requisitos de rendimiento y la escalabilidad. EC2 las instancias y los contenedores son buenas opciones para las bases de código heredadas, ya que requieren el menor cambio para trasladarse a la nube y se pueden usar las EC2 instancias para dedicar los recursos de una instancia de cómputo, mientras que los contenedores pueden facilitar la administración y la alta utilización. Las funciones sin servidor ofrecen el nivel más alto de abstracción, que se puede utilizar para definir código que solo se ejecute en respuesta a eventos, lo que puede reducir los costes.

Cliente de juegos

El cliente del juego representa el dispositivo de software y hardware que el jugador utiliza para jugar a un juego. El cliente del juego proporciona el software para traducir las entradas del jugador en mensajes que se envían a un servidor para su procesamiento, y es responsable de gestionar las respuestas entrantes del servidor y de mostrar los resultados, como gráficos, al jugador. En los juegos multijugador en red en tiempo real, el cliente del juego suele mantener una conexión de red persistente con un servidor de juego durante una sesión de juego para reducir la latencia de la red y minimizar el tiempo de procesamiento. Sin embargo, el cliente del juego también puede interactuar mediante REST con un servidor de juegos o servicios de backend.

Mensajería

Por lo general, hay tres categorías principales de mensajes en los juegos:

- Mensajes sobre la participación de los jugadores dirigidos a un usuario o grupo de usuarios específicos, como invitaciones a juegos o notificaciones push
- Mensajes grupales entre jugadores, como el chat del juego
- `service-to-service` Mensajería S, como los mensajes JSON que se utilizan para integrar dos o más aplicaciones

Una estrategia común para enviar y recibir este tipo de mensajes es utilizar patrones de arquitectura de procesamiento asíncrono y publicador-suscriptor. AWS ofrece varios servicios que pueden ayudarte a implementar la mensajería en tu juego.

- Amazon Simple Notification Service (SNS): servicio gestionado para entregar mensajes entre editores y suscriptores mediante un patrón de arquitectura. `pub/sub` Los editores envían mensajes mediante una API a Amazon SNS, que entrega los mensajes de forma asíncrona a las aplicaciones suscritas y puede enviar notificaciones push directamente a clientes móviles o de escritorio con soporte para algunos de los servicios de notificaciones push más utilizados. Amazon SNS se puede utilizar para notificaciones push a los clientes, así como para casos de uso `service-to-service` de mensajería.
- Amazon Simple Queue Service (SQS): un servicio de colas totalmente gestionado que facilita la integración de los servidores de juegos con tu juego, independientemente del lenguaje de programación utilizado en cada uno de ellos. Muchas tareas de los juegos se pueden disociar y gestionar en segundo plano, como actualizar una tabla de clasificación o los valores del tiempo

de juego en una base de datos. Este enfoque es una forma eficaz de desvincular varias partes del juego y escalar de forma independiente las funciones orientadas al jugador a partir del procesamiento interno.

- Amazon Managed Streaming for Apache Kafka (MSK): un servicio totalmente gestionado que simplifica la creación de aplicaciones de streaming de datos y de productores o consumidores mediante Apache Kafka, una popular solución de código abierto. Kafka se suele utilizar para ingerir y procesar datos de streaming en tiempo real y se puede utilizar para enviar mensajes. service-to-service
- Amazon ElastiCache (Redis OSS): proporciona un almacén de datos en memoria totalmente gestionado que incluye compatibilidad con la popular pub/sub función de Redis, que se utiliza habitualmente para desarrollar aplicaciones de salas de chat y mensajería de alto rendimiento. service-to-service Redis también admite tipos de datos enriquecidos, como listas y conjuntos, de modo que los desarrolladores pueden utilizar Redis para hacer colas de alto rendimiento.
- Amazon Pinpoint: proporciona mensajes de participación de los usuarios a través de correo electrónico, SMS, voz y notificaciones push. Por ejemplo, Amazon Pinpoint se puede usar para enviar mensajes de participación de los usuarios a los jugadores para invitarlos a volver al juego y se puede usar para casos de uso transaccional, como admitir tokens de autenticación multifactorial, confirmaciones de pedidos y correos electrónicos de restablecimiento de contraseñas.

Operaciones de juego en directo (Live Ops)

Las operaciones de juego en vivo (Live Ops) son un estilo de gestión y operaciones de juego que trata el juego como un servicio en vivo y ofrece continuamente nuevas funciones, actualizaciones, promociones, eventos dentro del juego y mejoras en el juego lanzado para mejorar la experiencia de la comunidad de jugadores.

Tradicionalmente, los juegos se ofrecían como productos y no como servicios, y con frecuencia se incorporaban nuevos contenidos y funciones en versiones o secuelas posteriores en lugar de incorporarse al producto lanzado. Con el enfoque de gestión de juegos de Live Ops, un equipo de operaciones de juego puede lanzar un juego y mantener una comunidad de jugadores comprometida mediante la experimentación, las promociones, los eventos del juego y la innovación para entretener a los jugadores.

Si bien este enfoque tiene la ventaja de permitir nuevas estrategias de participación de los jugadores y generar fuentes de ingresos recurrentes, requiere más experiencia operativa. Por ejemplo, para

implementar una estrategia de Live Ops exitosa, es posible que un desarrollador necesite integrarse con los servicios en la nube u operar su propia infraestructura técnica de back-end. También necesitan una forma eficaz de identificar y responder a los problemas que surjan en el juego o en la comunidad de jugadores y que puedan afectar negativamente a la experiencia del jugador.

Excelencia operativa

El pilar de la excelencia operativa se centra en las mejores prácticas para implementar y operar juegos basados en la nube a gran escala. Es importante centrarse en la excelencia operativa para mantener una experiencia de jugador positiva e implementar medidas preventivas para prepararse y recuperarse de los problemas que afectan a su experiencia.

Áreas de interés

- [Principios de diseño](#)
- [Operaciones en directo](#)
- [Estructura de la cuenta](#)
- [Implementaciones de juegos](#)
- [Monitorización de la salud](#)
- [Prueba de carga](#)
- [Optimización a lo largo del tiempo](#)
- [Recursos](#)

Principios de diseño

Además de los principios de diseño del documento técnico de Well-Architected Framework, los siguientes principios de diseño pueden ayudarlo a lograr la excelencia operativa en la creación y operación de juegos:

- Defina objetivos mensurables y alcanzables para los equipos de operaciones del juego y adáptelos según sea necesario: debido a que los juegos se basan en los éxitos, es difícil determinar de antemano cuántos jugadores jugarán a su juego cuando se lance o qué expectativas tendrán los jugadores con respecto a sus operaciones de juego en curso. Es importante establecer objetivos ambiciosos pero alcanzables con las partes interesadas y diseñar un enfoque que pueda ampliarse si el juego supera las previsiones y reducirse mientras los equipos de desarrollo del juego optimizan la experiencia del jugador. Prepárate y pruébalo adecuadamente con antelación para cumplir con estos requisitos y alinear a las partes interesadas empresariales y técnicas con los objetivos establecidos para el funcionamiento del juego. Una vez definidos los objetivos, los equipos del juego pueden lograr un equilibrio adecuado entre coste y rendimiento durante la planificación, el diseño, el aprovisionamiento, las pruebas, el despliegue y el funcionamiento de la infraestructura de back-end del juego.

- Utilice los manuales operativos para planificar la ampliación de las actividades relacionadas con el lanzamiento de los juegos y los eventos especiales: los equipos de operaciones del juego deben coordinarse con las partes interesadas de la empresa para modelar las proyecciones con respecto al número máximo de jugadores previsto para los eventos y llevar a cabo una planificación proactiva para preescalar la capacidad de la infraestructura con antelación. Debido a la naturaleza fluctuante del tráfico de jugadores durante los eventos, la planificación previa y las actividades de preescalado deberían aumentar tus sistemas de escalado automatizado existentes para aumentar tus posibilidades de éxito durante un evento y comprobar que tienes suficientes recursos para ofrecer una experiencia de juego positiva. Implemente prácticas de ingeniería de rendimiento para desarrollar una base de referencia de sus recursos y una comprensión basada en datos de la capacidad de su sistema, lo que le ayudará a guiar las actividades de preescalado y las configuraciones de escalado automatizado. Desarrolle manuales operativos para proporcionar coherencia al proceso. Esta planificación anticipada y la capacidad de respuesta a la demanda de los jugadores son especialmente importantes para los juegos en directo, que deben mantener un rendimiento y una infraestructura fiables para mantener una base de jugadores activa y comprometida durante un período prolongado.
- Establece un modelo operativo para recibir, investigar y responder a las solicitudes de asistencia de los jugadores: tras el lanzamiento, supervisa las denuncias de quejas y problemas relacionados con el juego. Implementa sistemas adecuados para interactuar con los jugadores de forma segura y eficaz a fin de resolver adecuadamente los problemas de los jugadores, como foros comunitarios, redes sociales, correo electrónico, sistemas de venta de entradas, centros de llamadas o soluciones de bots de chat automatizados. Esto es especialmente importante en el caso de los juegos en directo, que requieren una comunicación continua con la base de jugadores, una capacidad de respuesta a las opiniones de los jugadores para responder a las necesidades cambiantes y mantener una comunidad comprometida durante un período prolongado.

Operaciones en directo

GAMEOPS01: ¿Cómo defines la estrategia de operaciones en vivo (Live Ops) de tu juego?

Desarrolla una estrategia de operaciones en tiempo real (Live Ops) para tu juego basada en objetivos definidos y métricas de rendimiento, en consulta con las partes interesadas de la empresa.

Prácticas recomendadas

- [GAMEOPS01-BP01 Usa los objetivos del juego y las métricas de rendimiento empresarial para desarrollar tu estrategia de operaciones en vivo](#)

GAMEOPS01-BP01 Usa los objetivos del juego y las métricas de rendimiento empresarial para desarrollar tu estrategia de operaciones en vivo

Consulte a las partes interesadas de la empresa, como los productores de juegos y los socios de publicación, para determinar los objetivos y las métricas de rendimiento de un juego. Esto puede ayudarte a elaborar planes sobre la forma en que gestionarás el juego, incluida la definición de los plazos de mantenimiento, los programas de actualización del software y la infraestructura y los objetivos de fiabilidad y capacidad de recuperación del sistema.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Estas métricas también pueden ayudarte a determinar en qué fase del ciclo de vida del juego deberías incorporar un sistema operativo en tiempo real (Live Ops) para supervisar el estado del juego, recopilar comentarios directos sobre el juego y crear procesos de lanzamiento simplificados y automatizados. Por ejemplo, un juego nuevo puede esperar a alcanzar una escala determinada, medida por el número de jugadores activos, los ingresos u otro conjunto de indicadores, antes de formar un equipo dedicado a las operaciones en vivo. Es posible que un estudio de desarrollo de juegos establecido ya tenga experiencia de operaciones en directo, tal vez para sus otros juegos, por lo que solo tendría que incorporar el nuevo juego.

Pasos para la implementación

- Puedes definir los objetivos de concurrencia de jugadores (CCU) y de usuarios activos diarios y mensuales (DAU y MAU) que la infraestructura del juego debería poder soportar de manera efectiva, tus presupuestos de infraestructura, objetivos financieros y otros objetivos de rendimiento, como la frecuencia de publicación de contenido y funciones para aumentar la participación de los jugadores. Estos objetivos y métricas se utilizan para tomar decisiones sobre el diseño del juego, la gestión de las versiones, la observabilidad y el soporte que se necesitan para que las operaciones funcionen de forma eficiente.
- Es posible que tu juego tenga como objetivo publicar nuevas actualizaciones de contenido al menos una vez al mes sin que se produzca ningún tiempo de inactividad durante el lanzamiento.

Esta información te ayuda a definir tu estrategia de implementación de versiones y a coordinar la programación del mantenimiento necesario, que puede requerir tiempo de inactividad en otros momentos del mes, y a contribuir a tu SLA de disponibilidad.

Estructura de la cuenta

GAMEOPS02: ¿Cómo estructuraréis el alojamiento de vuestros Cuentas de AWS entornos de juego?

Implementa una estrategia multicuenta para aislar los diferentes entornos de juego y mejorar la seguridad, la eficiencia operativa y la escalabilidad. Úsalo AWS Organizations para administrar las jerarquías de cuentas, aplicar barreras a las cuentas y hacer cumplir las políticas de etiquetas y el etiquetado de los recursos desplegados.

Prácticas recomendadas

- [GAMEOPS02-BP01 Adopta una estrategia multicuenta para aislar diferentes juegos y aplicaciones en sus propias cuentas](#)
- [GAMEOPS02-BP02 Organice los recursos de infraestructura mediante el etiquetado de recursos](#)

GAMEOPS02-BP01 Adopta una estrategia multicuenta para aislar diferentes juegos y aplicaciones en sus propias cuentas

Diseñe una estructura contable que guíe la implementación de la infraestructura para cumplir con las necesidades operativas, de seguridad y de aislamiento de cada entorno. Es esencial aislar el entorno restringiendo el acceso a él y permitiendo que solo se utilicen AWS los servicios necesarios, ya que los entornos de producción están bloqueados y los entornos de desarrollo y pruebas son menos estrictos para permitir la experimentación. Se recomienda encarecidamente aislar aún más los principales subsistemas de cada entorno y los servicios comunes que utilizan varios entornos para alojarlos y gestionarlos por sí solos. Cuentas de AWS

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Adopte una estrategia multicuenta AWS aislando los diferentes entornos (como el de desarrollo, las pruebas, la puesta en escena, la producción y los servicios compartidos) de forma individual. Cuentas de AWS, lo que reduce el alcance de los incidentes. Considere AWS Organizations la posibilidad de gestionar de forma centralizada la jerarquía de sus operaciones. Cuentas de AWS para simplificar aún más sus operaciones, así como de definir y aplicar políticas a nivel de cuenta y de unidad organizativa (nivel OU) de forma selectiva. Al diseñar una unidad organizativa y una Cuenta de AWS estructura adecuadas que se ajusten a sus necesidades de flujo de trabajo de desarrollo y producción, podrá optimizar sus costes y mejorar la escalabilidad.

- Adopte una estrategia multicuenta: aisle los entornos para reducir el radio de incidencia y simplificar las operaciones.
- Uso AWS Organizations: administre las cuentas de forma jerárquica, aplique políticas y permita un gobierno centralizado.
- Planifique la escalabilidad: diseñe estructuras contables detalladas e implemente medidas de ahorro de costos para el crecimiento futuro.

Pasos para la implementación

Un sistema de juego implementado AWS debería utilizar varias cuentas que estén organizadas de forma lógica para proporcionar un aislamiento adecuado, lo que reduce el alcance de los problemas y simplifica las operaciones a medida que la infraestructura del juego se amplía. Cuentas de AWS que alojan la infraestructura del juego normalmente se agrupan en los siguientes entornos lógicos:

- Los desarrolladores utilizan los entornos de desarrollo de juegos para desarrollar el software y los sistemas del juego.
- Los entornos de prueba o control de calidad (QA) se utilizan para realizar pruebas de integración, controles de calidad manuales y otras pruebas automatizadas que deben realizarse.
- Los entornos de preparación o preproducción se utilizan para alojar el software completo, de modo que se puedan realizar pruebas de carga y humo antes de su puesta en producción.
- Los entornos en vivo o de producción se utilizan para alojar el software y la infraestructura en vivo y atender el tráfico de producción de los jugadores.
- Los entornos de herramientas o servicios compartidos proporcionan acceso a sistemas, software y herramientas comunes que utilizan muchos equipos diferentes. Por ejemplo, un repositorio central

de control de código fuente autohospedado y una granja de compilaciones de juegos pueden estar alojados en una cuenta de servicios compartidos.

- Los entornos de seguridad se utilizan para consolidar los registros centralizados y las tecnologías de seguridad que utilizan los equipos que se centran en la seguridad en la nube.

Para activar la infraestructura del juego AWS, se recomienda crear cuentas independientes para cada entorno de juego (desarrollo, pruebas, puesta en escena y producción), así como cuentas para la seguridad, el registro y los servicios compartidos centrales.

Por lo general, los estudios de desarrollo de juegos más pequeños que administran un número limitado de recursos de infraestructura (normalmente unos cientos de servidores o menos) pueden crear una Cuenta de AWS para cada uno de estos entornos (por ejemplo, una cuenta de producción, una cuenta de desarrollo y una cuenta de ensayo). Sin embargo, a medida que la infraestructura de juego o el tamaño del equipo aumentan con el tiempo, es posible que este modelo simplificado no se adapte bien.

Al configurar estos entornos, tenga en cuenta que muchos AWS servicios comparten los recursos y las [cuotas de servicio](#) a nivel de API para una cuenta completa dentro de una región en particular. Esto debe tenerse en cuenta a la hora de determinar cómo organizar las cuentas de forma lógica. Cuentas de AWS solo incurren en costos por consumir los servicios que se implementan en ellas. Por lo tanto, esto proporciona una forma de reducir de manera efectiva la contención de recursos y las cuotas de servicio, especialmente a medida que el juego crece y cada vez más desarrolladores necesitan acceso para crear y administrar los recursos.

Basándonos en nuestra experiencia trabajando con estudios de desarrollo de juegos más grandes, que suelen gestionar miles de servidores y cientos de desarrolladores acceden a los recursos, te recomendamos que diseñes una estructura contable más detallada en la que las aplicaciones individuales que respaldan tu juego tengan sus propias cuentas de desarrollo, pruebas, puesta en escena y producción. Como es difícil y lleva mucho tiempo rediseñar la estrategia de AWS cuentas múltiples después de lanzar el juego debido a la complejidad de la planificación y la migración de los sistemas activos, tenga en cuenta sus futuras necesidades de escalabilidad a la hora de determinar la estructura de cuentas múltiples adecuada.

[Se puede utilizar AWS Organizations para establecer una jerarquía y agrupamiento y definir las Cuentas de AWS unidades organizativas \(OUs\) para aplicarles políticas comunes a nivel de OU mediante políticas de control de servicios \(\)](#). SCPs AWS Organizations administra y gobierna su entorno de forma centralizada a medida que crece y escala sus recursos. Puede crear nuevas cuentas y asignar recursos mediante programación, agrupar cuentas para organizar sus flujos de

trabajo, aplicar políticas a las cuentas o grupos para su gestión y simplificar la facturación mediante el uso de un único método de pago para sus cuentas. Además, Organizations está integrado con otros servicios para que pueda definir configuraciones centrales, mecanismos de seguridad, requisitos de auditoría y uso compartido de recursos entre las cuentas de su organización.

[AWS Control Tower](#) proporciona una forma sencilla de configurar y gobernar un entorno seguro de múltiples cuentas, denominado landing zone. Control Tower crea tu landing zone utilizando la gestión y el gobierno continuos de las cuentas AWS Organizations, así como las mejores prácticas de implementación basadas en AWS la experiencia de trabajar con miles de clientes mientras se trasladan a la nube. [AWS Config](#), [AWS Trusted Advisor](#), y [AWS Security Hub CSPM](#) son servicios que proporcionan una visión agregada o centralizada del estado de salud de su cuenta.

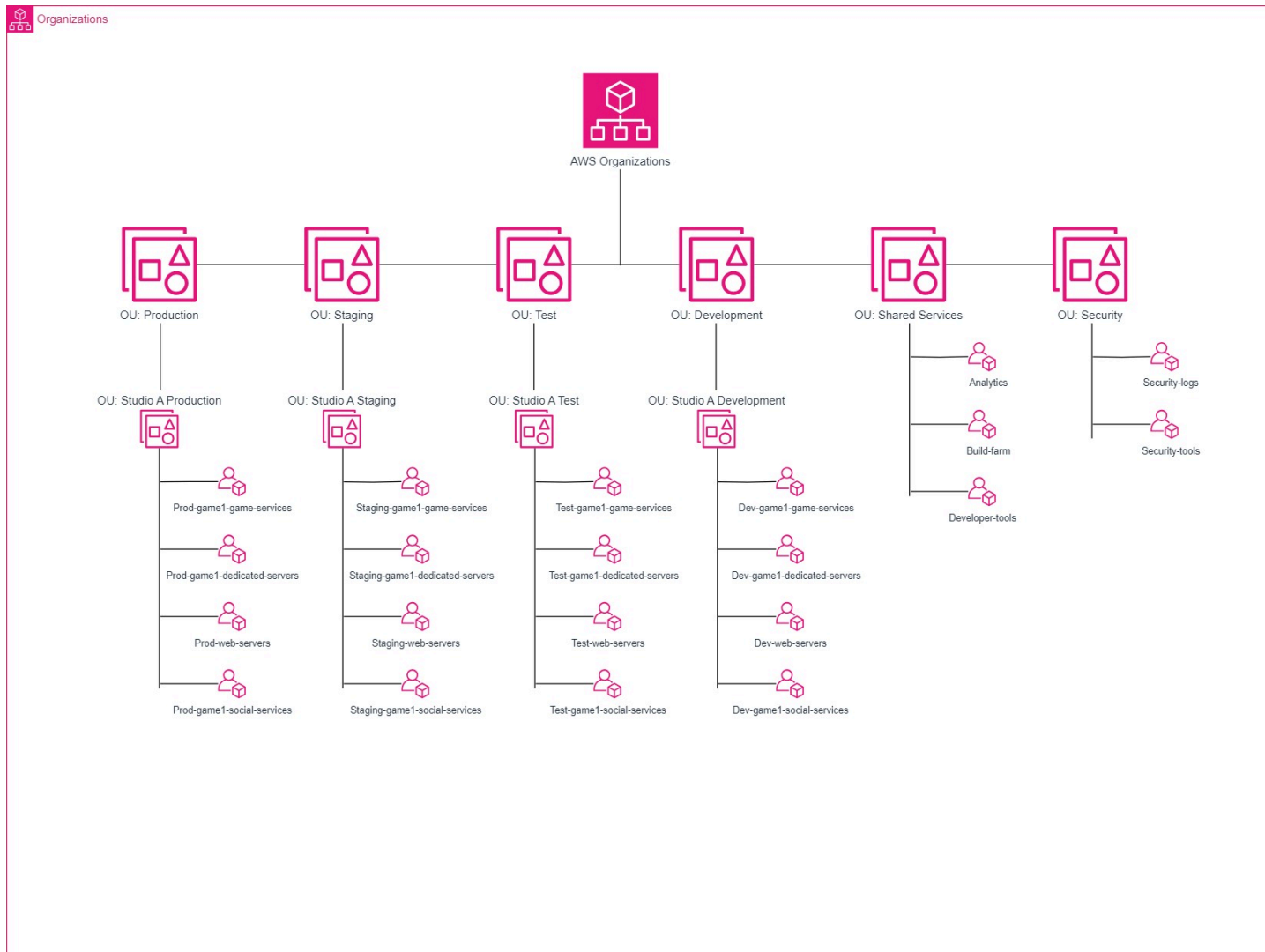
Este aislamiento te ayuda a configurar permisos y barreras personalizados o individuales para cada entorno de juego. Las cuentas de producción deben tener las barreras de protección, las restricciones de acceso, las herramientas de supervisión y alerta y de seguridad necesarias, mientras que las cuentas que no son de producción pueden no requerir el mismo nivel de barreras y permisos. Los entornos que no son de producción se pueden automatizar para cerrar los recursos fuera del horario laboral y ahorrar costes. La separación de las cuentas con este nivel de detalle facilita la supervisión de los costes de infraestructura de cada uno de los entornos en los que se basa un juego.

A continuación se muestra un ejemplo de una estructura de cuentas múltiples para una empresa de videojuegos que utiliza unidades AWS Organizations organizativas (OUs) para agruparse de forma lógica Cuentas de AWS en entornos y estudios independientes. En este ejemplo, se OUs utilizan para agrupar las cuentas en función de su entorno y, después, en función del estudio que gestiona el entorno. Esto demuestra cómo se puede crear una jerarquía anidada que permita implementar aplicaciones y juegos independientes en sus propias cuentas dentro de su entorno (se muestra como OUs), lo que puede resultar útil si se desarrollan y utilizan varios juegos. Consulte la documentación y los documentos técnicos que se proporcionan en la sección de recursos de este pilar para obtener información sobre otras estrategias que puede tener en cuenta para organizar su estrategia de cuentas múltiples.

Basándose en lo expuesto anteriormente, en el diagrama de ejemplo que aparece a continuación se parte del supuesto de que un estudio de videojuegos (organización) tiene un proceso de desarrollo compuesto por 4 etapas (desarrollo, pruebas, puesta en escena y producción). Para un juego determinado (game1), cada uno de los entornos (OU) tiene servicios de juego individuales Cuentas de AWS , servidores de juegos dedicados, servicios sociales y servidores web individuales. Los recursos que se ejecutan en cada uno Cuenta de AWS son relevantes para los subsistemas

respectivos. Por lo general, cada juego individual que utilice este tipo de proceso de desarrollo replicaría esta estructura o una similar para la suya Cuentas de AWS.

Además de este entorno centrado en el juego OUs, también existen las unidades organizativas de servicios compartidos y las unidades organizativas de seguridad. OUs Deberían ser aplicables a toda la organización, no a cada juego individual. De esta forma, los juegos consumirían los servicios compartidos de herramientas de desarrollo, datos y análisis, como en este ejemplo. A continuación, envíe los registros de la aplicación y del sistema a la Cuenta de AWS configuración para los registros de la unidad organizativa de seguridad.



Ejemplo de estructura de cuentas para entornos de juego

GAMEOPS02-BP02 Organice los recursos de infraestructura mediante el etiquetado de recursos

Para gestionar y realizar un seguimiento eficaz de [los recursos de su infraestructura](#) AWS, utilice el [etiquetado y la agrupación de recursos](#) adecuados para identificar el propietario, el proyecto, la aplicación, el centro de costes y otros datos de cada recurso. Los recursos etiquetados se pueden agrupar mediante [grupos de recursos](#), lo que contribuye al apoyo operativo.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Defina las [políticas de etiquetado](#). Las estrategias habituales incluyen etiquetas de recursos para identificar al propietario del recurso, como el nombre del equipo o el nombre de la persona, el nombre del juego, la aplicación o el proyecto, el nombre del estudio, el entorno (como el de desarrollo o producción) y la función del recurso (como el servidor de base de datos, el servidor web, el servidor de juegos dedicado, el servidor de aplicaciones o el servidor de caché). Puedes añadir otras etiquetas para ayudarte con las necesidades empresariales y de TI. [AWS Config](#) también puede aplicar una [política de etiquetado en el momento de](#) la creación y actualización de los recursos. Las etiquetas y los grupos de recursos están disponibles en las operaciones Consola de administración de AWS, AWS CLI, la y la API.

Pasos para la implementación

- Etiquete los recursos para identificar a su propietario, proyecto, aplicación, centro de costes y otros datos relevantes.
- Implemente políticas de etiquetado que incluyan etiquetas para el propietario, el proyecto, el estudio, el entorno y el rol del recurso.
- Úselo AWS Config para hacer cumplir las políticas de etiquetado y administrar las etiquetas a través Consola de administración de AWS de la CLI y la API.

Implementaciones de juegos

GAMEOPS03: ¿Cómo se gestionan los despliegues de juegos?

Gestiona los despliegues de juegos validando minuciosamente los componentes reutilizados, realizando tareas de ingeniería de rendimiento periódicas e implementando pruebas de carga periódicas durante todo el ciclo de vida del desarrollo.

Prácticas recomendadas

- [GAMEOPS03-BP01 Valida y prueba tu infraestructura y sistemas de juego principales existentes antes de reutilizarlos en tu juego](#)
- [GAMEOPS03-BP02 Realice ingeniería de rendimiento antes de cada lanzamiento \(o al menos para los lanzamientos principales\)](#)
- [GAMEOPS03-BP03 Prueba de carga temprana y con frecuencia](#)
- [GAMEOPS03-BP04 Adopta una estrategia de despliegue que minimice el impacto en los jugadores](#)
- [GAMEOPS03-BP05 Se requiere una infraestructura de preescalado para soportar los picos de demanda](#)

GAMEOPS03-BP01 Valida y prueba tu infraestructura y sistemas de juego principales existentes antes de reutilizarlos en tu juego

Las organizaciones tienden a reutilizar los componentes y el código fuente existentes de juegos anteriores para ahorrar tiempo y costes de desarrollo. Es posible que estos componentes y códigos heredados no estén sujetos a una revisión exhaustiva o a pruebas de integración detalladas y, en cambio, se basen en su rendimiento anterior.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Si bien la reutilización ayuda a mejorar la productividad, también puede implicar el riesgo de volver a introducir problemas de rendimiento y estabilidad anteriores en un nuevo proyecto. Por lo tanto, al reutilizar los componentes existentes y el código fuente de juegos anteriores, se deben implementar pruebas exhaustivas.

Pasos para la implementación

- Identifique el código y los componentes reutilizados: cataloga el código fuente, las bibliotecas y los componentes que se están reutilizando de juegos anteriores. Distinga claramente entre el código mantenido activamente y el obsoleto

- Documente el comportamiento original y los problemas conocidos: registre las características de rendimiento originales, las limitaciones funcionales y los errores o incidentes de producción conocidos relacionados con los componentes reutilizados.
- Realice una revisión exhaustiva del código: lleve a cabo una revisión técnica detallada de los componentes reutilizados, especialmente de aquellos que hayan tenido problemas en el pasado o que estén mal documentados.
- Reemplace o refactorice los componentes heredados de alto riesgo: priorice la sustitución o actualización de los componentes antiguos que tienen un historial de problemas o que ya no se pueden mantener, en lugar de recurrir a soluciones alternativas en la producción.
- Realiza pruebas de integración y compatibilidad: valida los componentes reutilizados en el contexto de los sistemas del nuevo juego. Comprueba que interactúan correctamente con los nuevos módulos, herramientas y APIs.

GAMEOPS03-BP02 Realice ingeniería de rendimiento antes de cada lanzamiento (o al menos para los lanzamientos principales)

La ingeniería del rendimiento es el proceso de supervisar varias métricas operativas clave de una aplicación para descubrir oportunidades de optimización que puedan mejorar aún más el rendimiento de la aplicación. Se trata de un proceso iterativo que comienza con las pruebas y, a continuación, con la optimización del código, sus dependencias, los procesos asociados, el sistema operativo anfitrión y la infraestructura subyacente.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Para realizar un análisis más profundo del rendimiento de la aplicación, integre una herramienta de supervisión del rendimiento de la aplicación (APM) o de depuración en el código de la aplicación que pueda aislar los problemas y reducir el tiempo de solución de problemas mediante el seguimiento de su comportamiento para detectar anomalías en los flujos de la aplicación. Las herramientas de APM también son capaces de identificar los métodos y las operaciones externas que funcionan con lentitud.

[AWS X-Ray](#) ayuda a los desarrolladores en sus actividades de ingeniería del rendimiento, como identificar los cuellos de botella en el rendimiento y analizar y corregir los errores de producción. Puede usar X-Ray para comprender el rendimiento de su aplicación y sus servicios subyacentes

e identificar y solucionar la causa raíz de los problemas y errores de rendimiento. A través de numerosas rondas de pruebas de carga, en las que la aplicación y su infraestructura se van cargando gradualmente con tráfico sintético de jugadores, se identifican diversos cuellos de botella en el sistema, errores de aplicación, excepciones, problemas del sistema operativo y otros problemas que quizás no se hayan detectado en otras pruebas de control de calidad.

Para eventos críticos como lanzamientos de juegos, lanzamientos de contenido, promociones y eventos importantes dentro del juego, usa [AWS Countdown](#), que proporciona una guía de implementación basada en manuales creados por expertos en juegos para verificar la preparación operativa, mitigar los posibles riesgos y planificar las necesidades de capacidad. AWS Countdown también tiene una opción de [soporte premium](#) que ofrece soporte mejorado y opciones, como ingenieros, para optimizar su infraestructura.

Pasos para la implementación

- La ingeniería del rendimiento implica evaluar y monitorear las métricas operativas clave para verificar que el código, los procesos, el sistema operativo y la infraestructura de la aplicación funcionen según lo esperado. La revisión previa a la producción también ayuda a definir el rendimiento de referencia en los diferentes niveles de uso simulado.
- Descubra métricas clave, como la utilización, los servicios, las E/S, los procesos, etc., y realice un seguimiento de ellas mediante herramientas del sistema como sar, top, vmstat, sysstat, netstat y Performance Monitor.
- Realice un seguimiento del rendimiento y el comportamiento de su aplicación mediante herramientas de APM, por ejemplo, AWS X-Ray para aislar problemas, identificar cuellos de botella y depurar errores de producción.
- Para eventos críticos, como el lanzamiento de juegos, suscríbase a AWS Countdown (IEM) para obtener orientación arquitectónica y operativa, soporte operativo a pedido y para identificar los riesgos y planificar las mitigaciones.

GAMEOPS03-BP03 Prueba de carga temprana y con frecuencia

Las pruebas de carga son el proceso de simular el tráfico real en un sistema para evaluar su fiabilidad y rendimiento.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Las pruebas de carga son un factor clave para desarrollar una base de rendimiento para sus recursos y comprender la capacidad del sistema, lo que puede servir de guía para la previsión financiera, el diseño de la arquitectura, la asignación de recursos, las configuraciones de escalado automatizadas y las actividades de preescalado posteriores al lanzamiento. Los beneficios adicionales incluyen:

- **Infraestructura optimizada:** los recursos pueden estar sobreaprovisionados o infrautilizados. Comprender los recursos necesarios se traducirá en costos más bajos y menos infraestructura que administrar.
- **Preparación para la escalabilidad:** Ciertos mecanismos y funciones pueden llevar a los usuarios a jugar rápidamente. Saber cuándo y cómo escalar puede marcar la diferencia entre satisfacer adecuadamente el aumento de la demanda o perder jugadores. Usa los resultados de las pruebas de carga para preparar manuales con los umbrales del sistema, los puntos de alerta y los puntos de alerta críticos en diferentes niveles de escalado.
- **Código de mayor calidad:** problemas como la interferencia excesiva entre los servicios, las llamadas no agrupadas a bases de datos, los algoritmos ineficientes, las pérdidas de memoria y los problemas de degradación de los servicios a veces son más fáciles de identificar a escala.
- **Validación del comportamiento:** al introducir diferentes tipos de errores en las pruebas, se puede validar el comportamiento esperado del sistema o descubrir problemas de gestión de errores que es necesario corregir.

Lo ideal es que los desarrolladores realicen las pruebas de carga en varios puntos del proceso de desarrollo, ya que cada uno de ellos puede ofrecer beneficios diferentes: desde el principio, guían las decisiones arquitectónicas y las iniciativas de refactorización, al tiempo que resulta más económico y sencillo realizar cambios. Al final de cada sprint o iteración, validan el rendimiento de la aplicación con las características y funcionalidades más recientes.

Antes de su implementación en producción, las pruebas de carga a gran escala que simulan los patrones de uso esperados en el mundo real confirman la capacidad del sistema para gestionar la carga de trabajo de producción. Tras la implementación, las pruebas de carga periódicas supervisan el rendimiento del sistema e identifican los cambios o los cuellos de botella que puedan surgir con el tiempo.

Para simular el tráfico de jugadores, necesitas clientes ligeros o bots que emulen los flujos de clientes del juego y realicen transacciones con el backend del juego para simular el comportamiento

de los jugadores en el mundo real. Por lo general, estos datos se recopilan mediante registros de jugadas y datos generados por pruebas de control de calidad realizadas por personas, así como mediante pruebas alfa o beta a escala limitada en el mundo real, en las que se invita a jugadores reales a jugar a una versión del juego de acceso anticipado.

Es importante registrar el comportamiento del sistema en un manual operativo para ayudar a solucionar posibles fallas en el futuro y conservar las métricas de rendimiento con las que se puedan comparar las pruebas de carga futuras. También se recomienda que el personal de control de calidad humano pruebe el juego mientras se está realizando la prueba de carga, ya que podrían descubrir problemas que los bots no puedan identificar y que las métricas no reflejen.

AWS El [servicio de inyección de fallos](#) es un servicio totalmente gestionado para realizar experimentos de inyección de fallos que facilitan la mejora del rendimiento, la observabilidad y la resiliencia de una aplicación. Los experimentos de inyección de errores se utilizan en la ingeniería del caos, que es la práctica de sobrecarga de una aplicación en entornos de prueba o producción mediante la creación de eventos disruptivos, como un aumento repentino del consumo de la memoria o la CPU, observando cómo responde el sistema e implementando mejoras. Los experimentos de inyección de errores ayudan a los equipos a crear las condiciones reales necesarias para descubrir los errores ocultos, monitorizar los puntos ciegos y los cuellos de botella en el rendimiento que son difíciles de encontrar en los sistemas distribuidos.

Pasos para la implementación

- Configura un entorno de pruebas de carga distribuido mediante [Guidance for Game Load Testing con base en Kubernetes](#).
- Personalice e implemente los módulos de control y de trabajo de Locust en el clúster de EKS utilizando los archivos de despliegue proporcionados, lo que permite una generación de carga escalable y gestionable.
- Registre el comportamiento y las métricas del sistema durante las pruebas de carga en un manual operativo para ayudar a resolver problemas en el futuro y establecer líneas base de rendimiento.
- Utilice experimentos de inyección de errores para simular las interrupciones del mundo real y descubrir problemas ocultos en el rendimiento, la observabilidad y la resiliencia del sistema.

GAMEOPS03-BP04 Adopta una estrategia de despliegue que minimice el impacto en los jugadores

Incorpore una estrategia de despliegue para el software y la infraestructura del juego que minimice el tiempo de inactividad que impide a los jugadores participar en el juego. Si bien algunos tipos de actualizaciones pueden requerir la instalación de nuevas actualizaciones en el cliente del juego, diseña el juego de forma que se minimice o evite la necesidad de que se produzcan tiempos de inactividad durante las implementaciones.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Uno de los pasos más importantes a tener en cuenta a la hora de desarrollar una estrategia de despliegue de juegos es determinar cómo se gestionará la infraestructura del juego. Gestiona la infraestructura de tu juego con una herramienta de infraestructura como código (IaC), como [AWS CloudFormation Terraform](#), de [Hashicorp](#), para reducir los errores humanos durante la preparación del entorno. Las plantillas de infraestructura se pueden implementar y probar en procesos automatizados, lo que crea coherencia en la configuración de los diferentes entornos de juego.

Existen varias estrategias de despliegue que se pueden utilizar en un juego:

Sustitución continua

El objetivo principal de una sustitución continua por el despliegue es realizar la versión sin cerrar el juego y sin afectar a los jugadores. Es importante que la actualización o los cambios que se vayan a realizar sean compatibles con versiones anteriores y funcionen de forma similar a las versiones anteriores del sistema.

En esta implementación, las instancias del servidor se sustituyen (sustituyen o implementan) de forma incremental por instancias que ejecutan la versión actualizada. Esta sustitución sucesiva se puede realizar de diferentes maneras. Por ejemplo, para implementar actualizaciones continuas en una flota de servidores de juegos dedicados, un enfoque típico consiste en crear un nuevo grupo de EC2 instancias de Auto Scaling que contenga la nueva versión de compilación del servidor de juegos implementada en ellos y, luego, dirigir gradualmente a los jugadores a las sesiones de juego alojadas en esta nueva flota de servidores. Si hay una actualización del cliente de juego asociada que es necesaria como requisito previo para usar la nueva versión del servidor de juegos, debes incluir una

comprobación de validación para comprobar que solo los jugadores que tengan instalada esta nueva actualización del cliente de juego puedan acceder a estas sesiones de juego.

Las flotas de servidores (por ejemplo, los grupos de EC2 Auto Scaling) que contienen la versión anterior de compilación del servidor de juegos solo se eliminan del servicio una vez agotadas las sesiones activas de los jugadores de forma adecuada, normalmente mediante la configuración de métricas de servidor individualizadas que permiten a los equipos de operaciones del juego automatizar este proceso. Como alternativa, para reducir la cantidad de infraestructura y el tiempo necesarios para realizar un despliegue continuo, se puede aplicar un enfoque alternativo en el que las instancias de producción existentes se retiran del servicio, se actualizan con la nueva versión del servidor de juegos y, a continuación, se vuelven a colocar en la flota de producción. Este enfoque reduce la cantidad de infraestructura necesaria, pero también aumenta el riesgo, ya que la cantidad de servidores de juegos en vivo disponibles para los jugadores se reduce a medida que se van sustituyendo los servidores.

Este modelo también se puede utilizar para realizar despliegues sucesivos en servicios de backend, como bases de datos, cachés y servidores de aplicaciones, que no alojan juegos. Siempre y cuando estos servicios se desplieguen de una manera altamente disponible con varias instancias agrupadas en clústeres, la complejidad de las implementaciones de estos servicios debería ser menor que la de las implementaciones en servidores de juegos dedicados.

Despliegue azul/verde

El objetivo principal de una blue/green implementación en un juego es minimizar el tiempo de inactividad y, al mismo tiempo, permitir una reversión segura a la implementación anterior si se detectan problemas. Es adecuado para despliegues en los que dos versiones del backend del juego sean compatibles y puedan servir a los jugadores simultáneamente.

En la estrategia de blue/green despliegue, se configuran dos entornos idénticos (azul y verde). La versión del juego existente está marcada en azul, mientras que la nueva versión del juego, que es el objetivo del despliegue, está etiquetada en verde. Cuando el entorno verde esté listo para la migración, puede configurar su capa de enrutamiento para transferir el tráfico al entorno verde y, al mismo tiempo, mantener el entorno anterior (azul) disponible en caso de que sea necesaria una conmutación por recuperación. En este escenario, las actualizaciones de enrutamiento pueden requerir la actualización del servicio de emparejamiento para configurarlo de modo que comience a enviar sesiones de juego a la nueva flota o, en el caso de los servicios de backend de juegos, esto podría consistir en actualizar los registros de DNS en Amazon Route 53 para tu servicio o [cambiar la ponderación del balanceador de carga de aplicaciones](#) para enviar tráfico a tu nuevo grupo objetivo.

Uno de los inconvenientes de la estrategia de blue/green implementación es el coste inherente del entorno de espera, debido a la infraestructura adicional que se requiere al realizar la implementación. Una opción para mitigar este costo adicional de infraestructura es considerar la posibilidad de adoptar una variante de blue/green implementación en la que el nuevo software de juegos se implemente en los mismos servidores que ya están implementados en la fase de producción. En este escenario, se puede iniciar un nuevo proceso de servidor ecológico con el nuevo software junto con el proceso de servidor azul existente, con la transición entre los procesos de servidor y no entre infraestructuras físicas independientes. Este enfoque también puede acelerar las implementaciones de juegos en una gran cantidad de infraestructura, ya que elimina la necesidad de esperar a que se lancen nuevos servidores en la nube. Para conocer las mejores prácticas sobre este enfoque de despliegue, consulte Implementaciones [azules/verdes](#) en AWS.

Implementación de Canary

La implementación de Canary es útil para los desarrolladores de juegos, ya que la estrategia se puede aplicar para lanzar una versión temprana de un juego en fase alfa o beta, o una función del juego, como un nuevo modo de juego, mapa o desafío, para un grupo reducido o reducido de jugadores en fase de producción. Este despliegue se denomina canario. Es posible que esta versión incluya funciones adicionales de seguimiento e informes, por lo que cuando jugadores reales juegan a ese juego o función, se recopila la telemetría de su jugabilidad y se analiza para detectar anomalías y problemas.

En el caso de las nuevas funciones, los jugadores no reciben notificaciones sistemáticas al respecto, y la telemetría del juego es la fuente principal que se utiliza para determinar si los jugadores tienen problemas, por lo que la versión debería retrasarse. Al mismo tiempo, si no se detecta ningún problema importante, la función podrá ampliarse a más jugadores para obtener información adicional. Si se notifica a los jugadores, se les puede pedir que envíen comentarios periódicos sobre su experiencia. Lo ideal sería que esta actividad de prueba fuera coordinada por un equipo de operaciones en vivo.

Como estrategia, el uso de Canary Deployment también se puede utilizar en versiones estándar para que los jugadores dispongan gradualmente de una nueva función. Una ventaja potencial con respecto al blue/green entorno estándar es que no se requiere un segundo entorno a gran escala. La capacidad del nuevo entorno reducido determina el número de jugadores que se incorporarán a la nueva función. Antes de añadir más jugadores, la capacidad debe escalarse adecuadamente. Si bien se espera que esta blue/green técnica personalizada cueste comparativamente menos que la técnica azul/verde estándar, se estima que tendrá un costo que puede ser superior al de la técnica de sustitución rodante, que se utiliza para los despliegues canarios.

Ejecute solo un canario en un entorno de producción y concéntrelo en sus datos y comentarios. Si se implementan varios canarios, se complica la solución de problemas y se aíslan los problemas de producción, y se perjudica la calidad de los conjuntos de datos y los comentarios que se recopilan.

Una variante del método canario es cuando uno o más experimentos (normalmente pruebas de interfaz de usuario) se realizan mediante despliegues específicos, en los que un conjunto de servidores back-end del juego sirve para una versión de una función y otro conjunto del mismo tamaño sirve para otra versión de la misma función. No se crea ninguna infraestructura adicional o especial para ello, y solo los grupos de servidores back-end elegidos reciben estas actualizaciones. El resultado de los experimentos consiste en observar cómo reaccionan los jugadores ante cada una de las versiones de la misma función, determinar si existe un consenso general sobre si les gusta o no les gusta y observar si hay problemas relacionados con su usabilidad o funcionalidad. Estos experimentos estratégicos también se denominan A/B pruebas, y el proceso general se denomina pruebas A/B. Al finalizar estos experimentos, se recopilan los datos de prueba necesarios antes de volver a la versión actual del sistema de backend del juego en los servidores utilizados para las pruebas.

Implementaciones tradicionales heredadas

En el estilo de despliegue tradicional, durante un período de mantenimiento programado, el juego se cierra y los jugadores conectados se eliminan o se agotan antes de que las instancias del servidor del backend del juego se actualicen con las versiones de código más recientes. Esta implementación afecta a los jugadores cada vez que se lleva a cabo, y hay que avisarlos antes de lo previsto. Como resultado, este modelo es el que más impacto tiene en los jugadores y debe evitarse siempre que sea posible.

Una vez implementada la actualización del juego, se puede hacer una prueba de humo antes de que los jugadores puedan acceder a él, quienes estarán esperando a que se vuelva a abrir el juego. Esto puede provocar un aumento de tráfico cuando los jugadores intenten iniciar sesión y jugar en poco tiempo. Por lo tanto, si el juego no está diseñado para soportar esos picos de tráfico, puedes optar por permitir que los jugadores regresen gradualmente al juego por lotes.

También puedes optar por aprovisionar en exceso la infraestructura para soportar el pico de tráfico inicial y, una vez que el tráfico del juego se estabilice, podrás reducir los recursos. Si es necesario, realiza este tipo de despliegue durante las horas de menor actividad, cuando el número de jugadores es más bajo. El mantenimiento programado con frecuencia, así como el mantenimiento prolongado, conllevan un riesgo inherente de pérdida de jugadores y una posible pérdida de ingresos. Los jugadores también esperan cambios tras una nueva versión y pueden perder la confianza en el juego una vez que regresen tras un periodo de inactividad.

Pasos para la implementación

- Minimice el tiempo de inactividad: implemente estrategias de despliegue que reduzcan el tiempo de inactividad y mantengan a los jugadores en el juego.
- Infraestructura como código (IaC): usa herramientas como AWS CloudFormation Terraform para administrar la infraestructura del juego y reducir los errores humanos.
- Estrategias de despliegue: usa uno o una combinación de despliegues de sustitución sucesiva, azul/verde y canario para ofrecer actualizaciones fluidas y reducir el impacto en los jugadores.

GAMEOPS03-BP05 Se requiere una infraestructura de preescalado para soportar los picos de demanda

Amplíe la infraestructura antes de los eventos de juego a gran escala para asegurarse de que puede hacer frente al repentino aumento de la demanda de jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Además de los lanzamientos de nuevos juegos, los juegos en directo suelen incluir eventos dentro del juego, promociones, contenido nuevo y lanzamientos de temporada como ejemplos de formas de mantener y mejorar la participación de los jugadores. Estas actividades generan un gran volumen de tráfico de jugadores durante la duración del evento o la promoción. La empresa espera alcanzar o superar los objetivos previstos para el evento, y la infraestructura del juego debe sostenerlos y apoyarlos durante todo este proceso.

Prepara tu infraestructura con antelación para que pueda soportar la cantidad de jugadores prevista que se producirá durante los eventos a gran escala. Para prepararse, los equipos de operaciones del juego deben coordinarse con las partes interesadas en ventas y marketing para estimar la demanda proyectada que se generará en un próximo evento teniendo en cuenta la concurrencia de jugadores anteriores, las métricas de participación y los datos de ventas. Si el evento es para el lanzamiento de un nuevo juego, los equipos de operaciones del juego deberían trabajar con estas partes interesadas para identificar proyecciones realistas a la escala que anticipan. Si bien puede resultar difícil predecir el éxito de un juego, es importante que todos entiendan cuáles son las expectativas de éxito para que la infraestructura pueda ampliarse y probarse para cumplir con esos objetivos.

Muchos juegos se lanzan por etapas, empezando por un lanzamiento gradual, que consiste en abrir el juego a un número reducido de jugadores y, después, ir escalando el número de jugadores

de forma orgánica en cada fase, antes de lanzarlos al público. Durante el periodo de lanzamiento provisional, supervisa, identifica, rastrea y resuelve los problemas y, al mismo tiempo, afina tus proyecciones para el lanzamiento público.

Para calcular correctamente los requisitos de infraestructura, recopila datos mediante pruebas de carga y rendimiento realizadas con los backends de tus juegos que se ejecutan en un entorno de producción o en un entorno de pruebas similar al de producción antes del lanzamiento del juego. Deberás realizar varias rondas de estas pruebas para simular diferentes condiciones del juego y comprobar que el backend puede soportar la carga en la mayoría de las condiciones.

Para lograrlo, los desarrolladores pueden crear robots de juego que recorran varios flujos de trabajo del juego y emulen diferentes condiciones. Estas pruebas deberían inspeccionar las diferentes capas del sistema del backend del juego para probar cada capa y cada componente y registrar los detalles. Usa los datos recopilados en estas pruebas para planificar el lanzamiento del juego.

Los puntos únicos de fallo (SPOF) deben identificarse y eliminarse siempre que sea posible, haciendo que la aplicación tenga una alta disponibilidad y sea tolerante a los errores. Utilice las pruebas de carga para identificarlos SPOFs emulando los fallos en diferentes niveles ascendentes y descendentes y verificando el comportamiento del juego y de otros componentes.

Configura el sistema para que se adapte automáticamente a la demanda, junto con la infraestructura estimada necesaria para el lanzamiento del juego, el evento o los preparativos de la promoción. Defina, configure y supervise los umbrales de los eventos de escalado para permitir que el backend del juego se escale y pueda soportar un gran volumen de tráfico de jugadores. Para el tráfico variable, lo mejor es el aprovisionamiento previo, ya que es posible que no haya tiempo suficiente para ampliarlo. Es posible que se requiera un escalado manual durante los lanzamientos iniciales de los juegos, ya que generan una demanda superior a la prevista con mayor rapidez de lo que los sistemas automatizados pueden escalar los recursos.

Sí AWS, las organizaciones deberían solicitar [cuotas de servicio](#) más altas para los servicios que utilizan en el backend del juego. Service Quotas se configuran para las cuentas a fin de evitar que los clientes instalen o escalen más infraestructura de la prevista de forma inadvertida. Cuando un juego que se ejecuta en una cuenta alcanza el límite superior de la cuota de servicio configurada en esa región, el servicio limita las solicitudes por encima de la cuota aprovisionada e interrumpe las provisiones. Los límites pueden provocar errores imprevistos o imprevistos y afectar a la experiencia del jugador. Supervisa, monitorea y revisa periódicamente los límites de las cuotas de servicio de los servicios utilizados por el juego durante la fase de producción para evitar que se produzcan limitaciones. Cuando el uso supera un límite de cuota de servicio tolerable, se puede solicitar un

aumento de la cuota planteando un [caso de soporte](#) desde el Centro de soporte de la consola, tras iniciar sesión en la cuenta afectada o mediante la [API de soporte](#).

[Para eventos críticos, como el lanzamiento de juegos, el lanzamiento de contenido, las promociones y los principales eventos del juego, usa AWS Countdown.](#) Countdown proporciona una guía de implementación basada en manuales creados por expertos en juegos para garantizar la preparación operativa, mitigar los posibles riesgos y planificar las necesidades de capacidad. AWS Countdown también cuenta con una opción de [soporte premium](#) que ofrece soporte mejorado y opciones, como ingenieros, para optimizar tu infraestructura.

Si vas a lanzar un juego alojado en Amazon GameLift, revisa las [listas de verificación previas al lanzamiento](#) para prepararte.

Pasos para la implementación

- Amplíe la infraestructura con antelación: prepare la infraestructura con antelación para los eventos de juegos a gran escala a fin de hacer frente a los repentinos aumentos de la demanda de jugadores.
- Calcula la demanda: coordínate con el departamento de ventas y marketing para estimar la demanda proyectada utilizando datos de jugadores anteriores y proyecciones realistas.
- Pruebas de carga y eliminación del SPOF: realice varias rondas de pruebas de carga para validar la capacidad del backend, identificar los puntos únicos de fallo y configurar correctamente el escalado automatizado.

Monitorización de la salud

GAMEOPS04: ¿Cómo se monitorea el estado del juego?

Supervisa el estado del juego implementando una instrumentación integral para detectar y rastrear los problemas que afectan a los jugadores, incluido el registro de la actividad del lado del cliente, la supervisión del servicio de back-end y la notificación de errores. Usa una combinación de AWS herramientas AWS X-Ray, como Amazon, CloudWatch y soluciones de terceros para identificar y resolver problemas rápidamente.

Prácticas recomendadas

- [GAMESOPS04-BP01 Instrumenta el juego para detectar y monitorear los problemas que afectan a los jugadores](#)

GAMESOPS04-BP01 Instrumenta el juego para detectar y monitorear los problemas que afectan a los jugadores

Además de responder a las redes sociales y a las denuncias de problemas por parte de los jugadores, utiliza soluciones de monitorización para detectar e investigar los problemas que puedan afectar a los jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Ninguna cantidad de pruebas puede identificar todos los problemas de un juego. Por lo general, los juegos se lanzan con problemas conocidos que se prevé que se solucionen gradualmente con la próxima versión del juego. Los problemas conocidos y reproducibles son fáciles de abordar y corregir. Para ayudar a identificar estos problemas, los clientes de juegos deberían implementar el seguimiento de la actividad de los jugadores, el registro de aplicaciones y la generación de informes en varios lugares estratégicos para ayudar al equipo de backend a identificar los problemas del lado del cliente. La capacidad de detectar estos problemas a tiempo ayuda a los desarrolladores de juegos a solucionarlos y solucionarlos antes de que se generalice. Los datos y registros incluidos en el código de seguimiento nunca deben incluir información de identificación personal (PII) y solo deben contener metadatos específicos del juego que ayuden a la depuración.

Implementa una solución de observabilidad para detectar problemas como errores o fallos en el juego y responder a ellos. Puedes usar [Amazon CloudWatch Synthetics](#) para crear canarios que puedan monitorear el estado de tus servicios de juegos de back-end orientados a los jugadores. Puedes configurar tus servicios de backend [AWS X-Ray](#) para rastrear las solicitudes en los servicios distribuidos y enviar tus registros y métricas personalizados a [Amazon CloudWatch](#).

Las soluciones de terceros, como [Backtrace.io](#) y [Sentry](#), son soluciones populares para informar de errores en los juegos. [También son populares las soluciones de monitoreo del rendimiento de las aplicaciones \(APM\) de socios como New Relic, Splunk, Datadog y Honeycomb.io.](#)

El equipo de operaciones en vivo del juego y los administradores de la comunidad también deberían monitorear varias redes sociales y canales para comprobar si hay comentarios de los jugadores, quejas e informes de errores, además de los canales de soporte oficiales. Revisa e intenta reproducir todas las quejas específicas del juego o envíalas al equipo de control de calidad para que las revise.

Si es reproducible, envía el problema a los desarrolladores del juego para que lo solucionen y lo solucionen antes de que afecte a una base de jugadores más amplia.

Pasos para la implementación

- Implemente soluciones de monitoreo: utilice herramientas de monitoreo para detectar problemas que afecten a los jugadores y responder rápidamente.
- Realice un seguimiento de la actividad y los registros de los jugadores: los clientes de juegos de instrumentos registran la actividad de los jugadores y notifican problemas, además de comprobar que no incluyen información de identificación personal (PII).
- Usa AWS herramientas y herramientas de terceros: usa herramientas como CloudWatch X-Ray y soluciones de terceros para informar de errores y monitorear el rendimiento, y monitorea las redes sociales para ver los comentarios de los jugadores y los informes de errores.

Prueba de carga

GAMEOPS05: ¿Qué debes tener en cuenta a la hora de realizar pruebas de carga de un juego?

Al realizar una prueba de carga de un juego, ten en cuenta la fase de prueba, la arquitectura generadora de carga y el marco de pruebas adecuados para evaluar de forma eficaz el rendimiento y la escalabilidad del sistema. Elige la combinación adecuada de tiempo (desarrollo inicial, sprints, preproducción o posterior al despliegue), infraestructura (EKSEC2, Fargate o Lambda) y herramienta de pruebas (JMeterLocust, Grafana K6 o Gatling) para adaptarla a las características únicas y los objetivos de desarrollo de tu juego.

Prácticas recomendadas

- [GAMEOPS05-BP01 Elija el escenario, la arquitectura y el marco de pruebas de carga adecuados para cumplir sus objetivos](#)

GAMEOPS05-BP01 Elija el escenario, la arquitectura y el marco de pruebas de carga adecuados para cumplir sus objetivos

El enfoque de las pruebas de carga de un juego puede variar considerablemente en función de muchos factores, como la fase del proceso de desarrollo en la que se lleve a cabo, la arquitectura

del propio sistema generador de carga y la elección del marco de pruebas de carga. El momento en que se lleve a cabo, ya sea en las fases iniciales, durante los sprints iterativos, antes de la implementación en producción o después de la implementación, determinará los objetivos y el enfoque de las pruebas. Los diferentes diseños de infraestructura generadora de carga tienen sus propias ventajas y desventajas, y la selección del marco de pruebas de carga influye en gran medida en las capacidades, la facilidad de uso y las integraciones disponibles para el proceso de prueba. Al alinear cuidadosamente estos elementos, los equipos de desarrollo pueden adaptar el enfoque de las pruebas de carga a las características únicas del juego, extraer la información más valiosa sobre el rendimiento y ofrecer una experiencia fluida a sus jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Pruebas de carga en diferentes etapas de desarrollo

La realización de pruebas de carga exploratorias al principio de las fases de desarrollo puede validar la arquitectura del sistema subyacente. Esto ayuda a los desarrolladores a tomar decisiones informadas sobre la infraestructura del juego, el diseño de la base de datos y la topología de la red antes de realizar un extenso trabajo de implementación. Las pruebas de carga identifican los riesgos y crean una base de rendimiento, lo que podría minimizar la necesidad de costosas modificaciones y la deuda técnica más adelante en el ciclo de vida del desarrollo. También pueden fomentar una comprensión compartida de los requisitos de rendimiento del juego entre el equipo, lo que se traduce en una mejor colaboración y toma de decisiones. En última instancia, las pruebas de carga durante las fases iniciales sientan una base sólida para un juego de alto rendimiento, escalable y resistente, lo que ayuda a mejorar la experiencia general del jugador.

Al final de cada sprint o iteración, las pruebas de carga permiten evaluar el impacto en el rendimiento de las nuevas funciones, las correcciones de errores y otros cambios introducidos en el último ciclo. Este enfoque específico permite a los equipos de desarrollo identificar rápidamente las regresiones o degradaciones del rendimiento introducidas por las últimas actualizaciones, lo que les permite abordar estos problemas antes de que se propaguen más adelante y mantener un nivel constante de calidad y rendimiento.

Antes de pasar a la fase de producción, las rigurosas pruebas de carga ayudan a los equipos a validar la capacidad del sistema para hacer frente a las condiciones de tráfico y carga previstas en el mundo real. Pueden detectar los problemas de escalabilidad o las limitaciones de recursos en la infraestructura de producción y ofrecen la oportunidad de optimizar el rendimiento del juego, creando una experiencia de usuario fluida y con gran capacidad de respuesta desde el primer día.

La información obtenida a partir de las pruebas de carga previas al lanzamiento puede mitigar los riesgos del día del lanzamiento y servir de base para la planificación continua de la capacidad, lo que sienta las bases para la sostenibilidad y la escalabilidad a largo plazo del juego.

Las pruebas de carga de un juego que ya está en fase de producción permiten a los equipos supervisar el rendimiento del juego e identificar las regresiones o degradaciones del rendimiento que puedan producirse con el paso del tiempo. Esto les permite abordar los problemas de forma proactiva antes de que afecten a la experiencia del jugador y afecten negativamente a la retención de los usuarios. Además, las pruebas de carga en la producción validan la eficacia de los esfuerzos de optimización del rendimiento o el escalado de la infraestructura que se han implementado. Este proceso proporciona a los jugadores una experiencia de juego de alta calidad, con capacidad de respuesta y escalable, incluso a medida que el juego evoluciona y madura.

Arquitecturas generadoras de carga

El diseño de la arquitectura generadora de carga para las pruebas de carga de los juegos puede adoptar diversas formas, cada una con su propio conjunto de ventajas y consideraciones.

En el nivel más básico, las EC2 instancias de [Amazon](#) autogestionadas se pueden aprovisionar y configurar para que actúen como generadores de carga. Con un enfoque de nodos de control y nodos de trabajo, puede configurar varias instancias generadoras de carga, cada una de las cuales ejecute su propio script de prueba y, en general, se administre mediante una única instancia de control. La arquitectura puede ampliarse y generar más carga sin aumentar la complejidad mediante la creación de nodos de trabajo adicionales, pero este enfoque práctico requiere que los equipos se encarguen del aprovisionamiento, la configuración y la administración de la infraestructura subyacente.

Para un enfoque más escalable y organizado, puede utilizar los clústeres de Kubernetes de [Amazon EKS](#) para gestionar y distribuir la carga de trabajo de pruebas de carga entre una flota de agentes de carga basados en contenedores. Las funciones de escalado automático de Kubernetes se pueden utilizar para gestionar el escalado de los pods generadores de carga, mientras que los propios equipos configuran y administran las instancias subyacentes del clúster que aloja los pods. EC2

Como alternativa, la naturaleza sin servidor [AWS Fargate](#) puede acelerar y simplificar la configuración de las pruebas de carga, ya que permite prescindir de la administración de la infraestructura y, al mismo tiempo, ofrecer la escalabilidad y la flexibilidad necesarias. En el caso de las soluciones híbridas en las que ya existe un clúster de Kubernetes local que genere carga, pero puede que se necesite capacidad adicional, [EKS Anywhere](#) puede gestionar ambos clústeres como uno solo desde el. Consola de administración de AWS

También puede utilizar [AWS Lambda](#) funciones en función de sus requisitos y objetivos. Las funciones Lambda son relativamente sencillas de configurar y escalar sin necesidad de aprovisionar ni administrar recursos adicionales. También permiten crear escenarios de prueba más complejos y dinámicos gracias a su profunda integración con otros AWS servicios. Sin embargo, las funciones Lambda tienen límites en cuanto a las funciones simultáneas y al tiempo de ejecución (15 minutos), lo que puede limitar la escala y la duración de las pruebas de carga que se pueden realizar. Las latencias de arranque en frío también pueden afectar a la precisión de los resultados, y las limitaciones de recursos de Lambda pueden no ser adecuadas para cargas de trabajo de pruebas de carga muy exigentes.

Los estudios que deseen utilizar una solución prediseñada pueden utilizar [Distributed Load Testing on AWS](#). Esta solución utiliza Amazon ECS AWS Fargate para implementar contenedores que pueden ejecutar simulaciones de decenas de miles de usuarios conectados. Puede usarlo para iniciar rápidamente su infraestructura de pruebas de carga al estilo IAC utilizando AWS CloudFormation.

Marcos de pruebas de carga

No hay dos marcos de pruebas de carga creados de la misma manera. Algunos tienen interfaces gráficas intuitivas para la creación de pruebas, mientras que otros se basan completamente en la línea de comandos. Una herramienta puede ser flexible y eficaz, pero su configuración y administración requieren tiempo y esfuerzo, y otra puede no tener servidor pero tener limitaciones en cuanto a las pruebas que puede crear y ejecutar. Algunas disfrutan de grandes comunidades y de abundantes tutoriales, y no están probadas sobre el terreno, lo que contrasta marcadamente con otras, que pueden haber sido sometidas a duras pruebas en producción, pero que carecen del apoyo o la documentación de la comunidad. Elige el marco que logre el equilibrio adecuado para ti y tu equipo. Algunas de las opciones más populares son:

- [Apache JMeter](#): popular marco de pruebas de carga de código abierto y basado en Java debido a su sólido conjunto de funciones y a su facilidad de uso. Su capacidad para simular escenarios de usuario complejos, su amplia gama de protocolos compatibles, sus informes exhaustivos y su historial comprobado lo convierten en JMeter una opción fiable para las pruebas de carga.
- [Locust](#): marco de pruebas de carga moderno y distribuido basado en una arquitectura basada en eventos, lo que lo hace eficiente y, al mismo tiempo, ahorra recursos. Las pruebas están escritas en Python, lo que permite escenarios de prueba flexibles que aprovechan miles de potentes bibliotecas de terceros, sin dejar de ser fáciles de leer y fáciles de leer.
- [Grafana K6](#): potente marco de pruebas de carga que combina la facilidad de uso con capacidades avanzadas. Su soporte para la generación de carga distribuida, la creación de scripts flexible y la

perfecta integración con Grafana para la visualización de datos hacen de Grafana K6 una opción atractiva.

- [Gatling](#): marco de pruebas de carga de código abierto conocido por su rendimiento y escalabilidad. Su lenguaje de dominio específico (DSL) basado en Scala permite a los desarrolladores crear scripts de pruebas de carga concisos y fáciles de mantener, y sus sólidas capacidades de generación de informes y análisis proporcionan información detallada del sistema que se está probando.

Pasos para la implementación

- Etapas de las pruebas de carga: realice pruebas de carga en varias etapas de desarrollo (desarrollo inicial, sprints, preproducción y posterior a la implementación) para validar el rendimiento del sistema e identificar los problemas.
- Arquitecturas generadoras de carga: elija las arquitecturas generadoras de carga adecuadas (EC2EKS, Fargate o Lambda) en función de las necesidades de escalabilidad, las preferencias de administración y los requisitos de prueba específicos.
- Marcos de pruebas de carga: seleccione un marco de pruebas de carga (como JMeter Locust, Grafana K6 o Gatling) que equilibre la facilidad de uso, el rendimiento, la flexibilidad y el apoyo de la comunidad para adaptarse a las necesidades de su equipo.

Optimización a lo largo del tiempo

GAMEOPS06: ¿Cómo optimizas tu juego a lo largo del tiempo?

Optimice su juego a lo largo del tiempo supervisando las métricas clave y los datos de telemetría para identificar las tendencias de los jugadores, el rendimiento del sistema y las áreas de mejora. Actualiza continuamente el diseño, la infraestructura y el enfoque de las pruebas de carga del juego en función de estos datos y, al mismo tiempo, adáptate a las nuevas tecnologías y marcos para ofrecer un rendimiento y una experiencia de juego óptimos a medida que el juego evoluciona.

Prácticas recomendadas

- [GAMEOPS06-BP01 Supervisa las métricas clave del juego para identificar las tendencias y patrones de los jugadores y utiliza la información para mejorar el juego](#)

- [GAMEOPS06-BP02 Actualiza y adapta el enfoque de pruebas de carga a medida que cambia el juego](#)

GAMEOPS06-BP01 Supervisa las métricas clave del juego para identificar las tendencias y patrones de los jugadores y utiliza la información para mejorar el juego

Además del uso del sistema cliente del juego, el uso de las aplicaciones, las excepciones y los datos sobre bloqueos, capture los datos de telemetría del juego que se envían al sistema back-end del juego. Estos datos deben representar la actividad de los jugadores para que puedas entender cómo interactúan los jugadores con las distintas funciones del juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Según su implementación, los clientes del juego pueden recopilar datos de telemetría en funciones predefinidas del juego o en ubicaciones del mundo del juego. Los datos se envían al servicio de ingestión interno para su procesamiento. Si no se puede acceder al servicio de backend, los clientes pueden almacenar los datos localmente en el dispositivo local hasta que el servicio de backend vuelva a estar disponible. Los diseñadores del juego utilizan estos datos de telemetría para revisar cómo juegan los jugadores y si hay anomalías en el juego.

Por ejemplo, los movimientos de los jugadores y las interacciones con los elementos de un mapa pueden extraerse de los datos de telemetría y trazarse como un mapa térmico de las actividades de los jugadores en el juego durante un período de tiempo determinado. Estos datos ayudan a los diseñadores del juego a identificar la necesidad de equilibrar varios elementos del juego, como el poder de un arma, el poder de un personaje del juego o la complejidad de un mapa. Por lo general, los datos de telemetría sin procesar se almacenan y, a continuación, se procesan para extraer análisis que puedan visualizar los analistas.

La implementación de la AnalyticsPipeline solución [Game](#) ayuda a los desarrolladores de juegos a lanzar una canalización de datos escalable y sin servidor para ingerir, almacenar y analizar los datos de telemetría generados a partir de juegos y servicios. La solución admite la ingesta de datos en streaming, lo que permite a los usuarios obtener información sobre sus juegos y otras aplicaciones en cuestión de minutos.

Para la ingesta, el almacenamiento, el procesamiento y el análisis de datos de telemetría de juegos personalizados, AWS también ofrece una serie de [servicios especializados para el procesamiento y el análisis de macrodatos](#).

Pasos para la implementación

- Capture datos de telemetría del juego: recopile datos sobre la actividad de los jugadores, el uso del sistema, las excepciones y los bloqueos para comprender las interacciones de los jugadores e identificar problemas.
- Implemente la recopilación de datos telemétricos: utilice funciones o ubicaciones predefinidas del juego para recopilar datos de telemetría y enviarlos a los servicios de backend, almacenándolos localmente si no se puede acceder a ellos.
- Utilice soluciones de AWS análisis: utilice AWS servicios como Game Analytics Pipeline para incorporar, almacenar y analizar datos de forma escalable, así como servicios especializados de procesamiento y análisis de macrodatos.

GAMEOPS06-BP02 Actualiza y adapta el enfoque de pruebas de carga a medida que cambia el juego

La optimización del enfoque de las pruebas de carga es un proceso continuo que debería evolucionar junto con el ciclo de desarrollo del juego. A medida que el juego crece en complejidad, base de usuarios y conjunto de funciones, la estrategia de pruebas de carga debe adaptarse para comprobar que simula con precisión las condiciones del mundo real y proporciona información útil.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Considere lo siguiente:

Escenarios de prueba inexistentes o desactualizados

A medida que se añadan nuevas funciones a un juego durante el proceso de desarrollo, crea y ejecuta nuevos escenarios de pruebas de carga para validar el rendimiento y la escalabilidad de las nuevas funciones. Del mismo modo, las características y la funcionalidad suelen rediseñarse para mejorar el rendimiento, tener en cuenta los comentarios de los jugadores o adaptarse a los nuevos objetivos de diseño, por lo que es necesario actualizar continuamente los escenarios de prueba para adaptarse a los cambios y comprobar y reflejar realmente el estado del sistema.

Nuevos marcos de pruebas de carga

Es posible que los desarrolladores necesiten cambiar los marcos de pruebas de carga por varios motivos:

- Es posible que el marco inicial ya no pueda simular adecuadamente la carga de usuarios ni proporcionar el nivel necesario de información sobre el rendimiento del sistema
- Es posible que las nuevas funciones del juego requieran la compatibilidad con pruebas de carga para nuevos protocolos o puntos de integración APIs
- Es posible que los desarrolladores deseen funciones más avanzadas a medida que se sientan más cómodos con el proceso de pruebas de carga
- Prefieren marcos que se ajusten mejor a la experiencia técnica, los lenguajes de programación o las cadenas de herramientas existentes del equipo

Al evaluarlo detenidamente y adaptarlo a lo largo del tiempo, los desarrolladores pueden adaptar el proceso de pruebas de carga a los requisitos cambiantes del juego y seguir proporcionando la información necesaria para optimizar y mejorar la experiencia general del usuario.

Optimizar los costes

La facilidad y la comodidad de utilizar AWS los servicios gestionados pueden ser muy beneficiosas, especialmente en las primeras etapas de desarrollo. Estos servicios resumen la gestión de la infraestructura subyacente, lo que permite a los equipos configurar rápidamente su solución y centrarse únicamente en diseñar escenarios de pruebas de carga y analizar los resultados. Sin embargo, el uso de los servicios gestionados suele tener un coste mayor debido al valor y la comodidad adicionales que ofrecen, como el aprovisionamiento, la configuración y el mantenimiento de la infraestructura, además de ofrecer capacidades de alta disponibilidad, escalabilidad y supervisión.

A medida que los equipos van madurando y se van sintiendo más cómodos y confiados con su proceso de pruebas de carga, puede llegar el momento en que la autogestión de la infraestructura pueda suponer una optimización adicional y un ahorro de costes adicionales. Si bien este enfoque práctico aumenta la sobrecarga operativa, tener un control directo sobre los recursos informáticos, las configuraciones, los comportamientos de escalado y la utilización de los recursos puede abrir nuevas oportunidades para afinar y reducir los costes. Por ejemplo, podría tener sentido que los equipos comiencen su proceso de pruebas de carga con una arquitectura AWS Fargate sin servidor y, posteriormente, pasen a gestionar automáticamente los nodos subyacentes en un clúster de Amazon EKS.

Pasos para la implementación

- Actualice los escenarios de prueba: cree y actualice continuamente los escenarios de pruebas de carga para validar las nuevas características y las funcionalidades refactorizadas, y compruebe que reflejan el estado actual del juego.
- Evalúe los marcos de pruebas de carga: adáptese a los nuevos marcos según sea necesario para simular la carga de usuarios, admitir nuevos protocolos y adaptarse a la experiencia y las cadenas de herramientas del equipo.
- Optimice los costes: comience con los AWS servicios gestionados para mayor facilidad y comodidad y, a continuación, considere la posibilidad de gestionar automáticamente la infraestructura para ahorrar costes a medida que el equipo se familiarice con el proceso de pruebas de carga.

Recursos

Consulta los siguientes recursos para obtener más información sobre nuestras prácticas recomendadas relacionadas con la excelencia operativa.

Documentación y blogs

- [Mejores prácticas de arquitectura para Game Tech](#)
- [Administrar tu estudio de juegos en la primera AWS parte](#)
- [Administrar tu estudio de juegos en AWS pt. 2](#)
- [Administrar tu estudio de juegos en AWS pt. 3](#)
- [Establecer su AWS entorno de mejores prácticas](#)
- [estrategia multicuenta para tu zona de aterrizaje de Control Tower](#)
- [Canal de análisis de juegos](#)
- [Maximice su información sobre los datos de sus juegos con Game Analytics Pipeline](#)
- [Aprovechamiento AWS Glue de Amazon Redshift Spectrum para obtener información sobre los jugadores](#)
- [¿Cómo configurar una canalización en CI/CD](#)
- [Cómo Good Job Games acelera un 43% con AWS Build Pipeline](#)
- [Implementación de un proceso de creación para las aplicaciones móviles de Unity](#)

- [Otros CI/CD blogs relevantes](#)
- [Juega de forma DevOps fácil con el blog Game-Server CD Pipeline](#)
- [Harmony Games implementa un backend de juego totalmente personalizado que utiliza \(\) AWS Cloud Development Kit \(AWS CDK\)AWS CDK](#)
- [GameLiftPrepárate para el lanzamiento](#)
- [Alojamiento de servidores de juegos híbridos con Amazon Gamelift Anywhere](#)
- [Acelera el desarrollo de servidores de juegos con Amazon Gamelift Anywhere y Amazon Gamelift Agent](#)
- [Cómo alojar tu juego de Unreal Engine por menos de 1 dólar por jugador con Amazon Gamelift](#)
- [Nueva guía de soluciones para crear backends de juegos multiplataforma escalables en AWS](#)
- [Prueba el motor de juegos de back-end Pragma con 1 millón de usuarios simultáneos en AWS](#)
- [Cómo Code Wizards probó Nakama de Heroic Lab ante dos millones de jugadores simultáneos con AWS](#)
- [Mejores prácticas con las unidades organizativas](#)
- [AWS X-Ray](#)
- [AWS Cuenta regresiva](#)
- [AWS para Games Solutions Hub](#)

Soluciones de socios

- [New Relic](#)
- [APM de Splunk](#)
- [Backtrace.io](#)
- [Centinela](#)
- [API Datadog](#)
- [Honeycomb.io](#)

Documentos técnicos

- [Organizar su entorno mediante varias cuentas](#)
- [Introducción a los patrones de desarrollo de juegos escalables en AWS](#)

Videos

- [YouTubeserie: Building Games on AWS](#)
- [AWS para juegos: Boss LEVEL Podcast](#)
- [Re:Invent 2023: se extiende a AWS los primeros 10 millones de usuarios](#)
- [Re:Invent 2022: Cómo procesa Riot Games 20 TB de análisis diarios en AWS](#)
- [Re:Invent 2022: ¿Cómo crearon Riot Games un AWS motor de informes sobre gobernanza](#)
- [Re:Invent 2023: Implementación de patrones de diseño distribuido en AWS](#)
- [Re:Invent 2023: Con Mortal Kombat 1, podrás llegar a millones de jugadores](#)
- [Re:Invent 2022: La evolución de la ingeniería del caos en Netflix](#)
- [Re:Invent 2023: mejores prácticas para la gobernanza de la nube](#)
- [Re:Invent 2023: mejores prácticas para crear arquitecturas multirregionales en AWS](#)

Materiales de formación

- [Plan de estudios: introducción a AWS los juegos, parte 1](#)

Seguridad

El pilar de la seguridad incluye la capacidad de proteger la información, los sistemas y los activos y, al mismo tiempo, ofrecer valor empresarial mediante la evaluación y la mitigación de los riesgos. Debido a la visibilidad global y al gran número de jugadores, los juegos son un objetivo deseable para los explotadores, los piratas informáticos y otras personas que buscan formas de explotar y abusar de los sistemas. Esto a menudo puede resultar en una experiencia de juego decepcionante y en un aumento de los costes para el desarrollador del juego si no se cuenta con una sólida base de seguridad.

Como se describe en el [modelo de responsabilidad compartida](#), es importante entender qué aspectos de la seguridad son responsabilidad del cliente AWS y cuáles son responsabilidad del cliente para estar preparado para mantener una postura de seguridad sólida. Este pilar proporciona una guía sobre las mejores prácticas de seguridad en la nube que debe tener en cuenta al desarrollar y operar juegos en la nube.

Antes de diseñar un sistema, debe establecer un conjunto de prácticas recomendadas de seguridad que incluyan controles de acceso. Además, debe poder identificar los incidentes de seguridad y proteger sus sistemas y servicios, al tiempo que mantiene la confidencialidad e integridad de los datos mediante la protección de los datos. Debe tener un proceso ben definido y practicado para responder a los incidentes de seguridad. Estas herramientas y técnicas son importantes porque respaldan los objetivos empresariales, como la prevención de pérdidas financieras o el cumplimiento de las obligaciones reglamentarias.

Ejemplo de cliente

AnyCompany Games es un estudio de juegos ficticio que está en proceso de mejorar su postura de seguridad. La seguridad puede ser fácil de entender cuando hay una explicación de su aplicación directa. AnyCompany En esta sección, se utilizan los juegos para contextualizar las mejores prácticas de seguridad descritas en el pilar

Áreas de interés

- [Principios de diseño](#)
- [Aspectos básicos de seguridad](#)
- [Seguridad continua](#)
- [Identity and Access Management](#)

- [Control de acceso](#)
- [Detección](#)
- [Protección de la infraestructura](#)
- [Respuesta a incidentes](#)
- [Seguridad de las aplicaciones](#)
- [Automatice la seguridad](#)
- [Modelado de amenazas](#)
- [Recursos](#)

Principios de diseño

Además de los principios de diseño del pilar de seguridad del documento técnico Well-Architected Framework, los siguientes principios de diseño pueden reforzar la seguridad de la carga de trabajo de los juegos en la nube:

- Supervise y modere el comportamiento de uso de los jugadores: capture y analice los datos de uso para comprender cómo interactúan los jugadores con su juego y sus funciones sociales. Al analizar estos datos, puedes detectar y responder a los comportamientos abusivos e inapropiados que pueden degradar la experiencia de los jugadores.

Aspectos básicos de seguridad

GAMESEC01: ¿Cómo se implementan los fundamentos de seguridad para el desarrollo de juegos?

Los estudios de videojuegos requieren un enfoque de seguridad único que proteja tanto los entornos de desarrollo como los servicios de los jugadores en directo. Una estrategia de AWS seguridad sólida para los estudios de videojuegos requiere tres componentes interconectados: una estructura de múltiples cuentas, una autenticación sólida y una estrategia de autorización clara que utilice políticas de IAM. Una AWS estructura de cuentas múltiples permite a los estudios separar diferentes proyectos de juego, etapas de desarrollo y entornos de herramientas. Esto proporciona a los estudios un control más detallado sobre aspectos como el acceso a entornos o servicios específicos. Al

habilitar una autenticación sólida, los miembros del equipo pueden acceder de forma segura a los recursos de desarrollo, ya sea que trabajen en el estudio o de forma remota, al tiempo que mantienen controles estrictos sobre el código fuente, las versiones de juegos y las herramientas patentadas. Los estudios también deben tener una estrategia de autorización clara para conceder permisos utilizando el principio del mínimo privilegio en lo que respecta a los permisos y funciones de IAM. Utilice las funciones de IAM para asignar permisos entre las distintas funciones del equipo de desarrollo, por ejemplo, dando a los equipos de desarrollo acceso a AWS servicios de bajo nivel y restringiendo a los artistas y diseñadores a sistemas específicos de creación y gestión de activos. Este enfoque especializado verifica que los estudios de videojuegos puedan proteger su propiedad intelectual, mantener flujos de trabajo de desarrollo eficientes y escalar sus equipos de forma segura, al tiempo que proporciona a los desarrolladores el acceso adecuado para iterar rápidamente sus proyectos.

Prácticas recomendadas

- [GAMESEC01-BP01 Utilice funciones y acceso federado, en lugar del usuario raíz de la cuenta, para realizar acciones en su entorno AWS](#)
- [GAMESEC01-BP02 Se utiliza AWS Control Tower para configurar rápidamente un entorno de múltiples cuentas en AWS](#)
- [GAMESEC01-BP03 Utilice políticas de roles con privilegios mínimos que se adapten a funciones laborales específicas](#)
- [GAMESEC01-BP04 Utilice las funciones y las políticas de acceso federado junto con las políticas de acceso a nivel de cuenta para conceder el acceso a sus recursos AWS](#)
- [GAMESEC01-BP05 Usa un proveedor de identidad central](#)

GAMESEC01-BP01 Utilice funciones y acceso federado, en lugar del usuario raíz de la cuenta, para realizar acciones en su entorno AWS

Al crear una por primera vez Cuenta de AWS, se comienza con una identidad conocida como usuario raíz, a la que se accede mediante la dirección de correo electrónico y la contraseña asociadas a la cuenta. El usuario raíz tiene acceso completo a AWS los servicios y recursos de esa cuenta. En la mayoría de los casos, debe evitar utilizar el usuario root para day-to-day las tareas. Cuando se requiera el acceso desde la raíz, confirme que es absolutamente necesario y compruebe que existen registros y barreras adicionales para rastrear su uso.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

En una AWS Organizations configuración, cada cuenta sigue teniendo su propio usuario raíz, pero el day-to-day acceso debería gestionarse a través de los roles de IAM y los usuarios del Centro de Identidad de IAM. Crea un acceso basado en roles adaptado a las etapas y equipos del ciclo de vida de tu juego. Por ejemplo, es posible que el equipo de operaciones en vivo necesite permisos para gestionar los eventos del juego, mientras que los desarrolladores necesiten acceso a las actualizaciones automáticas. Cuando trabajes con servicios o socios de terceros, usa el acceso federado para permitir una colaboración segura sin exponer la infraestructura confidencial. Este enfoque verifica que cada usuario o socio solo tenga el acceso que necesita y, al mismo tiempo, mantiene la seguridad de la infraestructura del juego y de los datos de los jugadores.

Ejemplo de cliente

AnyCompany Los juegos implementaron un control de acceso basado en roles al desarrollar su nuevo juego. Al utilizar funciones de IAM específicas para sus diversos equipos de desarrollo, evitan el uso de credenciales compartidas. Esta configuración permite que un equipo de desarrollo asuma una función en los principales sistemas de juego, mientras que la función del equipo de contenido solo es capaz de acceder a los servicios de gestión de activos.

Pasos para la implementación

- No utilices el usuario root después de configurar una cuenta a menos que sea absolutamente necesario. Cree la cuenta, proteja al usuario raíz y cree inmediatamente las funciones de administración de IAM necesarias y asígnelas al usuario federado.
- Utilice el usuario raíz únicamente cuando necesite realizar [un número limitado de tareas que solo estén disponibles para el usuario raíz](#). Algunos ejemplos de estas tareas incluyen cambiar la dirección de correo electrónico del usuario raíz y cambiar el plan de AWS soporte.

GAMESEC01-BP02 Se utiliza AWS Control Tower para configurar rápidamente un entorno de múltiples cuentas en AWS

Si empiezas a usarlo AWS con una sola cuenta, es posible que tu estudio de juegos vaya desapareciendo a medida que avance el proceso de desarrollo del juego. Por ejemplo, con una sola Cuenta de AWS, podrías empezar a alcanzar los límites de servicio o los costes de distintos proyectos y cargas de trabajo podrían volverse más complejos. La creación de cuentas diferentes para distintos títulos y entornos de juegos permite a los equipos experimentar con nuevas funciones,

eludir los límites de servicio y mantener una postura de seguridad y cumplimiento normativo. Al implementar una estrategia de cuentas múltiples en AWS, puedes beneficiarte de la distribución de los límites de servicio entre varias cuentas y obtener información sobre tus AWS costes.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Es un error común pensar que usar varias unidades automáticamente Cuentas de AWS será más confuso y consumirá más tiempo. Por el contrario, el uso de AWS servicios diseñados para facilitar la gestión de varias cuentas puede ayudar a tu estudio de videojuegos a dedicar menos tiempo a gestionar tus cuentas.

Puedes utilizar AWS Control Tower este servicio para aprovisionar de forma segura un AWS entorno de múltiples cuentas. Se recomienda la Torre de Control si estás creando un nuevo AWS entorno, iniciando tu viaje en él AWS o si eres completamente nuevo en él AWS. Durante el breve proceso de configuración, puede integrarse con otros AWS servicios relacionados con la administración de las cuentas y el acceso de los usuarios AWS Organizations, como Service Catalog e AWS IAM Identity Center.

Ejemplo de cliente

AnyCompany Al principio, los juegos funcionaban desde un único dispositivo Cuenta de AWS, pero se toparon con varios obstáculos cuando uno de sus equipos de desarrollo alcanzó los límites de EC2 servicio durante una crucial prueba beta. Al mismo tiempo, su equipo de desarrollo de otro juego tuvo problemas para asignar recursos para su proceso de pruebas automatizadas. La situación llegó a un punto crítico cuando AnyCompany Games no pudo separar con precisión los costes entre los proyectos, lo que dificultó presupuestar el desarrollo de cada juego.

AnyCompany Luego, Games implementó una estrategia de cuentas múltiples utilizando AWS Control Tower. Crearon cuentas separadas para cada proyecto de juego, con entornos de desarrollo, control de calidad y producción distintos. Esta separación a nivel de cuentas aísla los datos y los activos de cada proyecto, por lo que los equipos que trabajan en un juego no pueden acceder a los recursos de otro ni modificarlos. De este AWS Organizations modo, establecieron una estructura de facturación centralizada que mostraba claramente los costes de infraestructura de cada juego y, además, crearon políticas de acceso para toda la organización.

Pasos para la implementación

- Utilice la torre AWS de control para configurar un entorno automatizado de múltiples cuentas.

- Organice las cuentas en función de los entornos (como el de desarrollo, control de calidad y producción).
- Utilice AWS IAM Identity Center y Service Catalog para centralizar los permisos de los usuarios y agilizar el aprovisionamiento de recursos en todas las cuentas.

GAMESEC01-BP03 Utilice políticas de roles con privilegios mínimos que se adapten a funciones laborales específicas

La configuración de las políticas de IAM es una parte esencial para establecer una base de seguridad sólida. Cuando establezca permisos con políticas de IAM, conceda solo los permisos necesarios para llevar a cabo una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Por ejemplo, los equipos de control de calidad necesitan acceso para cambiar las cosas en los entornos de prueba, pero no deben tener la capacidad de modificar el entorno de producción.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Puede comenzar con permisos amplios, como [políticas administradas](#), mientras explora los permisos necesarios para su carga de trabajo o caso de uso. A medida que su caso de uso vaya madurando, puede ir reduciendo los permisos que concede para alcanzar el objetivo de privilegio mínimo.

Pasos para la implementación

- Siga la práctica de los permisos con privilegios mínimos para crear funciones de IAM para los usuarios y las aplicaciones.
- Utilice políticas AWS gestionadas para ofrecer un acceso amplio con rapidez y, al mismo tiempo, identifique los permisos específicos que necesitan los equipos o las aplicaciones para realizar sus tareas.
- Los estudios también pueden utilizar la [generación de políticas de IAM mediante el analizador de acceso](#) para generar políticas de IAM personalizadas basadas en CloudTrail eventos que identifiquen las acciones y los servicios utilizados por una entrada de IAM.
- Revise periódicamente las políticas de IAM y edite las políticas excesivamente permisivas.

GAMESEC01-BP04 Utilice las funciones y las políticas de acceso federado junto con las políticas de acceso a nivel de cuenta para conceder el acceso a sus recursos AWS

AWS Los nuevos usuarios suelen utilizar las políticas de IAM solo cuando conceden acceso a otras personas. Sin embargo, si las utiliza AWS Organizations, considere cómo utilizar las políticas de control de servicios junto con las políticas de IAM para conceder a los miembros del equipo del estudio y a los contratistas los niveles de acceso necesarios.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Puedes crear políticas de IAM para permitir o denegar el acceso a AWS los servicios o a las acciones de la API compatibles con ellos. AWS Identity and Access Management Solo se pueden aplicar a las identidades de IAM, como los usuarios, los grupos o las funciones. Por ejemplo, se puede usar una política de IAM para proporcionar a un usuario acceso de solo lectura a Amazon S3.

Las políticas de control de servicios (SCPs) son una barrera para usted. Cuentas de AWS Un SCP no concede permisos, sino que se utiliza para restringir las acciones en los AWS servicios de las cuentas de los miembros individuales. Por ejemplo, un SCP puede denegar el acceso Cuenta de AWS a una región en particular.

Cuando se realiza una acción, la política de IAM pertinente se evalúa en combinación con la. SCPs Siguiendo con el ejemplo anterior, si un rol intenta ejecutar una EC2 instancia, IAM indica si está permitido («Permitir» para ec2:RunInstances) y SCPs determinará si su elección de región es válida (se permite «us-east-1», pero el SCP deniega «us-west-1»).

Agrupar las políticas de IAM y permite SCPs comprobar que cualquier persona que acceda a tus AWS recursos solo reciba los permisos adecuados que necesite. Es especialmente importante tener esto en cuenta si tus recursos Cuentas de AWS y tú se distribuyen en varias regiones, pero no todos los miembros de tu estudio de juegos necesitan acceder a todas ellas.

Puedes personalizar las políticas de IAM para conceder a equipos específicos permisos específicos para actualizar aspectos como las configuraciones de los juegos, gestionar los datos de los jugadores, configurar eventos promocionales y moderar el contenido generado por los usuarios. Mientras tanto, puedes utilizarlas SCPs para reforzar los controles de toda la organización, que son fundamentales para las operaciones del juego. Estas pueden incluir restringir el despliegue

únicamente a las regiones aprobadas en las que esté disponible el juego, ayudar a evitar el acceso no autorizado a los almacenes de datos confidenciales de los jugadores, hacer cumplir los requisitos de conformidad y controlar los costes limitando el uso de los servicios entre las cuentas de desarrollo.

Pasos para la implementación

- Usa las políticas de IAM para gestionar los permisos de usuarios, grupos o funciones individuales.
- Utilice las políticas de control de servicios (SCPs) AWS Organizations para aplicar los permisos a nivel de cuenta.
- Combine las políticas de IAM y SCPs conceda solo el acceso necesario a usuarios y cuentas específicos.

Recursos

- [Políticas y permisos en IAM AWS](#)
- [Políticas de control de servicios](#)
- [Políticas administradas de AWS para funciones de trabajo](#)

GAMESEC01-BP05 Usa un proveedor de identidad central

Un proveedor de identidad central actúa como una fuente única para almacenar y administrar las credenciales, las identidades, los permisos y la autenticación de los usuarios.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Utilice un proveedor de identidad central para agilizar el proceso de autenticación de los usuarios, aplicar políticas de seguridad coherentes y simplificar la administración de los usuarios en todas sus Cuentas de AWS aplicaciones. Tener un enfoque centralizado elimina la necesidad de administrar las identidades y credenciales de los usuarios por separado, lo que reduce el riesgo de inconsistencias, redundancias y otras vulnerabilidades de seguridad. La consolidación de las identidades y la autenticación de los usuarios en un solo lugar también permite mejorar la visibilidad, el control y la auditabilidad de todo el entorno. AWS

Ejemplo de cliente

AnyCompany Los juegos se enfrentaron a importantes desafíos a la hora de gestionar el acceso de los desarrolladores en su infraestructura en rápida expansión AWS . Su equipo de desarrollo pasó de 50 a 200 personas en tres títulos principales. Al principio, cada equipo de proyecto gestionaba sus propias credenciales de AWS acceso, lo que se traducía en prácticas de seguridad incoherentes, demoras en la incorporación de nuevos desarrolladores y ocasionales incidentes de seguridad.

El estudio implementó AWS IAM Identity Center como su proveedor de identidad central, consolidando la gestión de usuarios en un solo sistema. Lo conectaron con su directorio corporativo existente, lo que permitió a los desarrolladores utilizar las mismas credenciales de la empresa para AWS acceder. Ahora los desarrolladores utilizan el único nombre de usuario de su empresa para obtener el AWS acceso que necesitan para completar su trabajo

Pasos para la implementación

- Considere la posibilidad de utilizar el Centro de identidades de AWS IAM como su proveedor de identidad central. Esto proporciona una gestión de acceso uniforme en todas sus instalaciones Cuentas de AWS, proporciona a sus empleados una autenticación de inicio de sesión único y simplifica la auditoría del acceso de los usuarios a sus aplicaciones. AWS El IAM Identity Center también se conecta con las identidades corporativas existentes de los proveedores de identidad compatibles.

Seguridad continua

GAMESEC02: ¿Cómo se logran, mantienen y supervisan las mejores prácticas de seguridad continuas?

Cumplir con las mejores prácticas de seguridad es fundamental para las empresas de todos los sectores, pero especialmente en la industria del juego. La industria de los videojuegos se basa en cultivar y mantener la confianza de los jugadores y en crear una reputación sólida, e incluso los problemas de seguridad más pequeños pueden socavar rápidamente esa confianza.

Además, la naturaleza global de la industria del juego exige el cumplimiento de diversas normas y normas del sector que rigen la protección de datos, la privacidad de los consumidores y la seguridad en todas las regiones en las que se ofrecen los juegos. La jugabilidad justa y segura es otro aspecto fundamental que subraya la importancia de contar con medidas de seguridad sólidas. Hacer trampa, piratear y otras formas de explotación del juego pueden interrumpir la experiencia de juego de

los jugadores legítimos, por lo que es esencial contar con controles de seguridad estrictos para mantener la integridad del juego y fomentar la igualdad de condiciones para los participantes.

Prácticas recomendadas

- [GAMESEC02-BP01 Utilice plantillas listas para implementar para las prácticas de seguridad estándar](#)
- [GAMESEC02-BP02 Utiliza técnicas de reparación automatizadas cuando se produzca un incidente de seguridad](#)

GAMESEC02-BP01 Utilice plantillas listas para implementar para las prácticas de seguridad estándar

Ready-to-deploy las plantillas proporcionan una forma proactiva y ágil de evaluar su postura de seguridad en la nube. Las plantillas preconfiguradas evalúan la seguridad de la nube e implementan los cambios necesarios con prontitud. Las plantillas abarcan una amplia gama de mejores prácticas en diversas tecnologías y marcos de seguridad ampliamente aceptados. El uso de plantillas puede ayudar a los estudios de videojuegos a mantener configuraciones de infraestructura consistentes, especialmente porque pueden ampliarse y añadir más Cuentas de AWS para soportar nuevas cargas de trabajo.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Al utilizar AWS servicios e implementar ready-to-deploy plantillas, los desarrolladores de juegos pueden evaluar y reforzar de forma proactiva su estrategia de seguridad en la nube, salvaguardando su propiedad intelectual, protegiendo los datos de los jugadores y fomentando un entorno de juego seguro mediante evaluaciones de seguridad periódicas y una supervisión continua para identificar y abordar rápidamente las posibles vulnerabilidades.

Ejemplo de cliente

AnyCompany Los juegos se enfrentaron a un importante desafío a la hora de preparar el lanzamiento de su próximo título en la industria europea. Se dieron cuenta de que sus prácticas actuales de manejo de datos no cumplían con los requisitos del GDPR. Recurrieron a AWS Security Hub CSPM AWS Config y a sus ready-to-deploy plantillas en busca de una solución. El equipo implementó el paquete de conformidad específico del GDPR AWS Config, que evaluó automáticamente su infraestructura existente con respecto a los estándares del GDPR. Este análisis inicial reveló

varias lagunas críticas, como políticas de retención de datos inadecuadas y controles de acceso inadecuados sobre el lugar donde se almacenaban los datos de los jugadores. Utilizando las reglas predefinidas de la plantilla, AnyCompany Games implementó rápidamente los cambios necesarios. Además, las continuas comprobaciones de conformidad automatizadas que proporcionaba la plantilla permitieron al pequeño equipo mantener el cumplimiento del RGPD sin esfuerzo alguno, a la vez que continuaban actualizando y ampliando el juego.

Pasos para la implementación

- Utiliza plantillas para las prácticas de seguridad estándar, como las reglas gestionadas y los paquetes de conformidad incluidos AWS Config y estándares. AWS Security Hub CSPM
- Revisa los detalles de los [estándares CSPM de Security Hub](#) para determinar cuáles se ajustan mejor a las necesidades de seguridad de tu estudio de videojuegos.

GAMESEC02-BP02 Utiliza técnicas de reparación automatizadas cuando se produzca un incidente de seguridad

Al utilizar técnicas de reparación automatizadas, los desarrolladores de juegos pueden proteger y mantener su infraestructura de juegos de forma proactiva y minimizar el posible impacto que pueda tener un incidente de seguridad. Si se detecta un problema de seguridad, utiliza un manual para guiar tu respuesta a la situación. Automatice estas respuestas siempre que sea posible para solucionar los problemas con mayor rapidez y reducir su impacto. Esto mejora la experiencia del jugador al reducir la posibilidad de que se produzcan tiempos de inactividad e interrupciones en el juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Prepararse para responder a los problemas de seguridad no solo protege la experiencia de los jugadores, sino que también cumple con las diversas normas reglamentarias y de cumplimiento. Además, el uso de respuestas de seguridad automatizadas amplía las operaciones de seguridad a medida que aumentan las cargas de trabajo. AWS proporciona servicios para ayudar a identificar y automatizar la respuesta a estos incidentes.

Ejemplo de cliente

AnyCompany Los juegos se enfrentaron a un grave incidente de seguridad cuando, durante una actualización rutinaria de la cartera de activos, se hizo pública accidentalmente un depósito de

S3 que contenía modelos de personajes y texturas inéditos para su próximo juego. El sistema de seguridad automatizado detectó el cambio de permisos del depósito a los pocos minutos de producirse la modificación. El sistema ejecutó inmediatamente su manual de correcciones: devolvió el depósito al estado privado, registró los intentos de acceso durante el período de exposición, notificó al equipo de seguridad y creó un CloudTrail registro detallado de los cambios de permisos.

Pasos para la implementación

- Utilice la [respuesta de seguridad automatizada en](#) la AWS solución para implementar manuales de automatización que definan las acciones que se tomarán automáticamente en respuesta a los eventos de seguridad. AWS Security Hub CSPM

Recursos

- [AWS para el blog de juegos: Cómo gestionar tu estudio de juegos en AWS: primera parte](#)
- [AWS para el blog de juegos: cómo administrar tu estudio de juegos, AWS parte 2](#)
- [Registra una unidad organizativa existente con AWS Control Tower](#)
- [Cuenta de AWS usuario root](#)
- [Tareas que requieren credenciales de usuario root](#)
- [Respuesta de seguridad automatizada activada AWS](#)

Identity and Access Management

GAMESEC03: ¿Cómo se gestiona la identidad de los jugadores y la gestión de los accesos?

Al desarrollar un juego, debes determinar cómo proporcionarás a los jugadores acceso a tu juego y a los servicios relacionados. En la siguiente sección, se analizan las consideraciones de diseño, incluidas la autenticación, la autorización y la autenticación multifactorial de los jugadores.

Prácticas recomendadas

- [GAMESEC03-BP01 Determina tu enfoque para identificar y controlar el acceso de los jugadores al entorno y los recursos del juego.](#)
- [GAMESEC03-BP02 Autentica las solicitudes que se envían a tu servicio de backend de juegos](#)

- [GAMESEC03-BP03 Usa tu servicio de backend de juegos para validar las solicitudes de los jugadores para unirse a un juego multijugador](#)
- [GAMESEC03-BP04 Aplique una política de seguridad estricta para las cuentas de usuario de los jugadores exigiendo una contraseña segura](#)
- [GAMESEC03-BP05 Ofrece a los jugadores la opción de configurar la autenticación multifactorial \(MFA\) en sus cuentas](#)

GAMESEC03-BP01 Determina tu enfoque para identificar y controlar el acceso de los jugadores al entorno y los recursos del juego.

Esta decisión depende de tu estrategia de adquisición y monetización de jugadores, de la experiencia del jugador y de otros factores, como las capacidades actuales que puedan ofrecer tus socios editores de juegos. Por ejemplo, un juego puede requerir compras y requerir que el jugador cree un perfil de usuario para asociar métodos de pago con dinero real a su cuenta.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Como alternativa, un juego puede reducir las barreras de acceso a las experiencias de los jugadores primerizos, eliminando la necesidad de crear una cuenta de usuario antes de jugar, aumentando así las probabilidades de que un jugador pruebe el juego por primera vez. Por lo general, los juegos implementan una o más combinaciones de enfoques de gestión de la identidad y el acceso de los jugadores a sus juegos.

Acceso anónimo o no autenticado

Este nivel de acceso es útil en situaciones en las que un juego no requiere que el jugador cree una nueva cuenta de usuario o que se vincule con su identidad en las redes sociales y los sistemas de juego. Esta es la forma más sencilla y rápida para que un jugador comience a jugar y resulta especialmente útil en juegos para dispositivos móviles, en los que un desarrollador de juegos puede querer reducir la barrera de entrada para la experiencia inicial.

En este escenario de acceso, si quieres identificar el uso desde la instalación del juego, puedes programar el cliente del juego para que genere y almacene un identificador único en el dispositivo del jugador. Este identificador único se utiliza para identificar al jugador en todas las sesiones de juego en su dispositivo y permite elaborar informes analíticos sobre el uso a lo largo del tiempo. Más adelante, si un jugador decide crear una cuenta, podrás asociar su nueva cuenta de usuario a su

identificador único generado anteriormente. Esto vinculará su nueva identidad de jugador con su historial de uso, que puede incluir estadísticas y logros del juego.

Si un jugador finalmente no crea ni vincula una cuenta, el dispositivo que utiliza para interactuar con el juego se puede identificar de forma única, pero la información recuperable sobre el jugador no se recopila ni almacena. Por lo tanto, si el jugador rompe o pierde su dispositivo, los datos previamente almacenados asociados al dispositivo también se pierden y es posible que no se puedan recuperar.

Autenticación con nombre de usuario y contraseña

Un juego puede permitir a los jugadores crear sus propias cuentas de usuario con un nombre de usuario y una contraseña que se almacenan en el backend del juego. Esto puede ocurrir cuando un desarrollador de juegos colabora con un editor de juegos que ya tiene un sistema de cuentas de jugador con el que el desarrollador puede integrarse. Como alternativa, un desarrollador que publique sus propios juegos podría simplificar la experiencia de los jugadores permitiéndoles crear una cuenta de usuario única para acceder a todos los juegos que publiquen.

Autenticación y vinculación de cuentas con redes sociales y sistemas de juego de terceros

Es habitual que los juegos en línea y los juegos con funciones sociales proporcionen una federación de proveedores de identidad externos para simplificar la experiencia del jugador. En lugar de pedirles a los jugadores que creen una combinación de nombre de usuario y contraseña para autenticarse, puedes usar la federación de identidades para permitir que los jugadores se autenticuen con sus cuentas de terceros en redes sociales y sistemas de juego. Este proceso de inicio de sesión simplifica la experiencia de inicio de sesión y registro para los jugadores. También proporciona una alternativa práctica a la creación obligatoria de cuentas y un método sencillo para que los jugadores accedan a los juegos.

Para los desarrolladores de juegos, un proceso de inicio de sesión federado puede ofrecer un flujo de trabajo de verificación de jugadores simplificado. También puede proporcionar una forma más fiable de gestionar los datos de los jugadores que se utilizan para la personalización. Esto se debe a que no necesitas pedir a los jugadores que te proporcionen ciertos datos que probablemente ya hayan proporcionado al proveedor de identidad externo. Además, estos sistemas se integran con funciones sociales adicionales, como la posibilidad de vincular a los jugadores con sus amigos.

Pasos para la implementación

- Utiliza el acceso anónimo o no autenticado para reducir las barreras a las que se enfrentan los jugadores primerizos. Para ello, genera un identificador de dispositivo único para hacer un seguimiento del uso y permite vincular las cuentas más adelante.

- Implemente la autenticación por nombre de usuario y contraseña para las cuentas de usuario dedicadas, utilizando los sistemas de cuentas de jugadores existentes o creando una experiencia unificada en todos los juegos.
- Integre proveedores de identidad externos para la autenticación federada, simplificando los procesos de inicio de sesión y permitiendo el acceso a las funciones sociales y a los datos de personalización.

GAMESEC03-BP02 Autentica las solicitudes que se envían a tu servicio de backend de juegos

La autenticación de las solicitudes que se envían al servicio de backend de tu juego puede impedir que las solicitudes no deseadas se ejecuten correctamente.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Debes proporcionar un servicio de autenticación para que los jugadores inicien sesión, que devuelva al cliente del juego fichas seguras de corta duración, como un JSON Web Token (JWT), cuando un jugador se autentica correctamente.

Estos símbolos pueden incluir afirmaciones sobre las reclamaciones que contengan los atributos del jugador y otros metadatos relevantes. Estos metadatos relevantes se pueden usar en solicitudes posteriores que se envíen desde el cliente del juego al backend del juego para autenticar las solicitudes y autorizarlas en el contexto del jugador autenticado.

[Tiene la opción de diseñar y crear su propio sistema de autenticación de jugadores, lo que requeriría mejoras y mantenimiento continuos, o puede utilizar las funciones escalables y seguras de registro de usuarios, inicio de sesión y control de acceso que ofrece Amazon Cognito.](#)

Los grupos de usuarios de Amazon Cognito incluyen un directorio de usuarios para la autenticación y la autorización. Un grupo de usuarios permite APIs integrarlos en el juego para los flujos de trabajo de registro, inicio de sesión y restablecimiento de contraseñas, que se pueden integrar con proveedores de identidad de terceros. Tanto [Application Load Balancers](#) como [Amazon API Gateway](#) ofrecen integraciones con Cognito para integrar la autenticación de usuario en las solicitudes enviadas a sus backends de juegos personalizados alojados con estos servicios.

Si su juego admite el acceso anónimo y no puede autenticar a un jugador, puede utilizar un enfoque de autenticación de cliente para ofrecer una experiencia más segura al integrarlo con el backend del

juego. Si el cliente de tu juego usa AWS los servicios directamente, las solicitudes a estos servicios deben firmarse con credenciales. Para proporcionar credenciales al cliente del juego a los usuarios no autenticados, puede usar el AWS SDK para recuperar credenciales efímeras de los [grupos de identidades de Amazon Cognito que se pueden usar para](#) firmar sus solicitudes de servicios. AWS Estas credenciales se pueden actualizar desde el cliente del juego.

Además de integrarte directamente con el AWS SDK desde el cliente del juego, también puedes crear tu propio backend del juego mediante un servicio como [Amazon API Gateway](#), que admite la autorización personalizada. Al diseñar tu propio servicio de back-end de juegos, puedes obtener un control autoritario sobre las solicitudes con una lógica personalizada del lado del servidor.

Para obtener más información sobre cómo crear un servicio de back-end para juegos alojados en Amazon GameLift, consulta [Diseña tu servicio de cliente de juegos](#).

Ejemplo de cliente

AnyCompany Los juegos mejoraron la seguridad de su próximo título al adoptar un enfoque de autenticación y autorización gestionadas. En lugar de mantener un sistema personalizado de nombre de usuario y contraseña, utilizaron grupos de usuarios de Amazon Cognito para gestionar el registro e inicio de sesión de los jugadores, y grupos de identidades para permitir el acceso anónimo a los jugadores que probaban el modo de entrenamiento antes de crear una cuenta. También implementaron una lógica de autorización personalizada en el juego para reconocer las funciones de administrador definidas en Cognito, lo que permitió a los usuarios acceder a funciones de administración especiales del juego.

Pasos para la implementación

- Utilice los grupos de usuarios de Amazon Cognito para gestionar la autenticación con tokens seguros, por ejemplo JWTs, habilitando funciones como el registro, el inicio de sesión y el restablecimiento de contraseñas.
- Recupere credenciales efímeras de los grupos de identidades de Amazon Cognito para que los usuarios anónimos puedan interactuar de forma segura con AWS los servicios.
- Implemente backends de juegos personalizados mediante Amazon API Gateway para lograr una lógica de autenticación personalizada del lado del servidor.

GAMESEC03-BP03 Usa tu servicio de backend de juegos para validar las solicitudes de los jugadores para unirse a un juego multijugador

Por lo general, en los juegos multijugador, un jugador se une a una sesión de juego seleccionando una opción directamente de una lista de sesiones disponibles, o bien envía una solicitud para encontrar una pareja. Este último enfoque hace recaer en el desarrollador del juego la responsabilidad de localizar una sesión de juego válida y de proporcionar la información de conexión (normalmente una dirección IP y un número de puerto) al cliente del juego del jugador. La implementación puede variar en función del género de juego que estés desarrollando, pero en cualquier caso, una buena práctica de seguridad es validar desde el servidor la solicitud de un jugador para unirse a un juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Por ejemplo, en un juego multijugador basado en sesiones, la solicitud de un jugador para unirse a una sesión de juego debe ser validada por el software del servidor de juegos con el servicio de emparejamiento interno del juego antes de autorizar su conexión al servidor. Cuando un jugador solicita unirse a una sesión de juego, el servidor del juego debe comprobar la solicitud de un identificador único, como un identificador de sesión de jugador y un ticket generado por el servidor, que el servicio de emparejamiento interno del juego proporcionó previamente al cliente del juego.

Al iniciar la conexión con el servidor del juego, el software del servidor puede utilizar esta información para comprobar con el servicio de emparejamiento que la solicitud de conexión del jugador es válida y comprobar que el jugador no se une a una plaza que anteriormente estaba reservada en la sesión de juego para otro jugador.

Para los juegos que están alojados en Amazon GameLift, consulta [client/server Interacciones entre juegos con GameLift servidores de Amazon](#) para ver un ejemplo de cómo se puede implementar este tipo de validación del lado del servidor.

Ejemplo de cliente

Durante el lanzamiento de la versión beta inicial de AnyCompany Games, descubrieron que los jugadores se saltaban su sistema de matchmaking al conectarse directamente a los servidores del juego, lo que provocaba graves problemas de integridad competitiva. Cuando los jugadores mejor clasificados se dieron cuenta de que podían compartir las direcciones IP de sus servidores con sus amigos, empezaron a eludir el sistema de emparejamiento basado en habilidades, lo que provocó

que jugadores experimentados se unieran a partidas de novatos, lo que supuso una experiencia frustrante para los nuevos jugadores. AnyCompany Los juegos respondieron implementando un sistema de validación en el servidor que generaba tickets de sesión únicos para cada solicitud de emparejamiento. El sistema requería que tanto el jugador como el IDs emparejador solicitaran tickets y verificaran los intentos de conexión contra su servicio de emparejamiento interno.

Pasos para la implementación

- Valide las solicitudes de unión de los jugadores en el servidor mediante identificadores únicos, como la sesión del jugador y los tickets generados por el servidor. IDs
- Confirma la validez de las solicitudes de conexión con el servicio de búsqueda de parejas para bloquear el acceso no autorizado.
- Comprueba que jugadores no autorizados no accedan a las plazas reservadas en las sesiones de juego durante el proceso de validación.

GAMESEC03-BP04 Aplique una política de seguridad estricta para las cuentas de usuario de los jugadores exigiendo una contraseña segura

Si un juego ofrece a los jugadores la posibilidad de crear una cuenta de usuario con una contraseña, debes exigir que las contraseñas de los jugadores cumplan con políticas estrictas. Por ejemplo, los grupos de usuarios de Amazon Cognito le permiten [definir los requisitos de contraseña](#) para las cuentas de usuario. Si establece una política de contraseñas segura, puede evitar que las cuentas de sus jugadores se vean afectadas por la ingeniería social y los ataques de fuerza bruta.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Ejemplo de cliente

AnyCompany Los juegos se enfrentaron a una crisis cuando su popular título sufrió una oleada de secuestros de cuentas debido a una política de contraseñas poco rigurosa. Los jugadores que utilizaban contraseñas sencillas, como «password123», se estaban convirtiendo en víctimas de ataques automatizados de fuerza bruta, con el resultado de perder objetos y comprometer la moneda del juego. Para evitarlo, AnyCompany Games renovó su sistema de inicio de sesión y exigió que las contraseñas no se utilizaran anteriormente e incluyeran al menos una letra mayúscula, un número, un carácter especial y una longitud mínima de 15 caracteres.

Pasos para la implementación

- Exija políticas de contraseñas seguras para las cuentas de los jugadores a fin de mejorar la seguridad.
- Utilice los grupos de usuarios de Amazon Cognito para definir y aplicar los requisitos de contraseña.

GAMESEC03-BP05 Ofrece a los jugadores la opción de configurar la autenticación multifactorial (MFA) en sus cuentas

Las cuentas de los jugadores pueden ser una ventaja para los malos actores, especialmente en los juegos que admiten compras y divisas del juego. Debido a la omnipresencia del hackeo de las cuentas de los jugadores y los ataques de ingeniería social, ofrezca a los jugadores la opción de mejorar la seguridad de sus cuentas configurando la autenticación multifactor (MFA).

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Cuando un jugador intenta acceder a su cuenta mediante la MFA, se envía un código temporal a su dirección de correo electrónico, número de teléfono o a una aplicación móvil de autenticación multifactor diseñada específicamente. Para autenticarse correctamente, el jugador debe introducir el código en el sistema de inicio de sesión dentro de un período de tiempo limitado.

La MFA también se puede usar para ayudar a proteger las cuentas que intentan autenticarse desde una nueva ubicación geográfica, las cuentas que el soporte de los jugadores ha marcado como posibles actividades maliciosas e incluso las cuentas que no han iniciado sesión en el juego durante un período prolongado.

Por ejemplo, los grupos de usuarios de Amazon Cognito pueden [configurar la autenticación multifactor](#) en los directorios de usuarios.

Pasos para la implementación

- Habilita la autenticación multifactorial (MFA) para mejorar la seguridad de las cuentas de los jugadores.
- Usa códigos temporales enviados por correo electrónico, teléfono o aplicaciones de MFA para verificar el acceso a la cuenta.

- Aplica la MFA a nuevas ubicaciones geográficas, cuentas marcadas o cuentas con inactividad prolongada.

Control de acceso

GAMESEC04: ¿Cómo se bloquea el acceso no autorizado al contenido del juego?

Los juegos modernos incluyen una cantidad significativa de contenido, como el contenido descargable (DLC), que es un aspecto importante de la participación de los jugadores y la monetización de los juegos. Los jugadores esperan un flujo continuo de nuevos personajes, niveles y desafíos, lo que obliga a los desarrolladores de juegos a mantenerse al día con la demanda constante de contenido nuevo para retener a los jugadores. La variedad y el tamaño del contenido pueden variar considerablemente según el tipo de juego y el dispositivo en el que se juegue. Independientemente del sistema del juego, proteja el contenido del juego contra el acceso no autorizado.

Prácticas recomendadas

- [GAMESEC04-BP01 Restrinja el acceso al contenido descargable a clientes y usuarios autorizados](#)
- [GAMESEC04-BP02 Limite el acceso de origen a las redes de entrega de contenido autorizadas \(\)
CDNs](#)
- [GAMESEC04-BP03 Implemente restricciones geográficas para limitar el acceso no autorizado](#)
- [GAMESEC04-BP04 Restrinja el acceso al contenido con soluciones de gestión de derechos digitales \(DRM\)](#)

GAMESEC04-BP01 Restrinja el acceso al contenido descargable a clientes y usuarios autorizados

Restrinja el acceso al contenido del juego a las aplicaciones y los clientes autorizados. Considere la posibilidad de utilizar Amazon S3 como un origen rentable y escalable para almacenar contenido de juegos descargable y Amazon CloudFront para ofrecer a los jugadores contenido con rendimiento global. Ambos servicios proporcionan mecanismos integrados para restringir el acceso a los datos almacenados, como restringir el acceso a los usuarios autenticados.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Otorgar acceso al contenido almacenado en Amazon S3

Cuando necesite conceder acceso al contenido almacenado en S3, debe tener en cuenta varios factores. De forma predeterminada, solo la Cuenta de AWS persona que creó un bucket de S3 puede acceder a los objetos almacenados en él. Para conceder acceso a sus aplicaciones internas y gestionar el contenido almacenado en los buckets de Amazon S3, utilice [AWS Identity and Access Management \(IAM\)](#) para crear políticas que proporcionen el acceso adecuado.

Las [funciones de IAM](#) se pueden asociar a usuarios, sistemas o aplicaciones federados alojados en servicios, como Amazon EC2 AWS Lambda, y a aplicaciones basadas en contenedores alojadas en Amazon EKS y Amazon ECS. Por ejemplo, puede usar el AWS SDK o AWS CLI publicar y administrar activos de contenido de juegos en depósitos de S3. Para respaldar este caso de uso, puedes crear un rol de IAM con el acceso adecuado para leer y escribir contenido de juegos en tus buckets de S3 y asociarlo a las EC2 instancias que alojan tu software y tus scripts.

Se pueden definir políticas basadas en recursos para tu bucket y para objetos específicos. Las [políticas de bucket](#) de S3 están asociadas a un bucket de S3 y se pueden usar para restringir el acceso al bucket y a los objetos que contiene, así como para conceder acceso a sus recursos de Amazon S3 desde otras cuentas. Por ejemplo, en situaciones en las que varios equipos o estudios de desarrollo de juegos independientes trabajan en el mismo contenido de juego y requieren el mismo acceso al contenido alojado de forma centralizada en Amazon S3, puede utilizar una política de bucket de S3 para definir los permisos de acceso multicuenta a los recursos de S3. Considere la posibilidad de utilizar [los puntos de acceso de S3](#), que puede simplificar la administración del acceso a los datos compartidos mediante la creación de puntos de acceso con nombres y permisos específicos para cada aplicación o conjunto de aplicaciones. La documentación de Amazon S3 contiene [prácticas recomendadas adicionales para el control de acceso en Amazon S3](#).

Granting short-term access to your content

Cuando el acceso solo sea necesario durante un tiempo limitado y específico, genere datos temporales URLs que permitan el acceso a su contenido a corto plazo. Amazon S3 admite la generación de [prefirmados URLs](#), lo que permite a los propietarios de objetos conceder acceso por tiempo limitado a los objetos de Amazon S3 sin necesidad de actualizar la política de bucket. De este modo, el usuario final o la aplicación a los que se concede el acceso no necesitan tener una cuenta o permisos de IAM, sino que utilizan la URL prefirmada para acceder al contenido.

Esta es una práctica recomendada que se suele utilizar en diversos casos de uso de juegos, como conceder a los jugadores autorizados acceso al contenido descargable al que tienen derecho y proporcionar acceso temporal al contenido del juego por tiempo limitado. Los prefijos de URLs pueden usarse para proporcionar permisos temporales para subir contenido a un bucket de S3. Por ejemplo, podrías considerar la posibilidad de utilizar una URL prefijada para que un jugador pueda subir los registros de los clientes y así ayudar a tu equipo de asistencia a solucionar un caso de asistencia a un jugador.

Utilizar una red de distribución de contenido para proporcionar acceso a tu contenido

Si bien es posible que sus aplicaciones, desarrolladores de juegos, artistas y demás personal necesiten acceder directamente al contenido de los paquetes de S3 para fines de desarrollo y administración, utilice una red de entrega de contenido para proporcionar acceso al contenido que esté disponible públicamente para los jugadores u otros usuarios a través de Internet. Este enfoque mejora el rendimiento de las descargas y reduce los costes al almacenar en caché el contenido al que se accede con frecuencia. Amazon CloudFront puede distribuir tu contenido por todo el mundo almacenándolo en caché y entregándolo más cerca de tus jugadores, a la vez que reduce la carga en el origen de descarga del juego, como Amazon S3.

En lugar de publicar tu contenido público directamente desde los buckets de S3, se recomienda mantener este contenido privado y publicarlo públicamente mediante el uso de CloudFront. CloudFront se puede configurar para que los jugadores accedan a tu contenido privado (por ejemplo, la descarga de un juego nuevo solo para jugadores de pago) mediante [cookies firmadas URLs o firmadas](#). A continuación, puedes desarrollar tu aplicación para crear y distribuir los datos firmados URLs a los usuarios autenticados, o bien para enviar encabezados de cookies configurando cookies firmadas para los usuarios autenticados. Al crear cookies firmadas URLs o firmadas para controlar el acceso a los archivos, puede especificar una fecha y una hora de finalización, tras las cuales la URL y las cookies dejan de ser válidas.

Si lo prefieres, también puedes especificar la dirección IP o el rango de direcciones de los ordenadores que se pueden utilizar para acceder a tu contenido, lo que resulta útil si quieres restringir el acceso a redes de contratistas o estudios de desarrollo de juegos específicos. Utiliza cookies firmadas cuando quieras proporcionar acceso a varios archivos restringidos o si no quieres cambiar los actuales URLs. Utilízalas firmadas URLs cuando quieras restringir el acceso a archivos individuales o si tus usuarios utilizan un cliente que no admite cookies. Las cookies firmadas URLs tienen prioridad sobre las cookies firmadas.

Pasos para la implementación

- Utilice las funciones de IAM y las políticas de compartimentos para conceder el acceso adecuado a los depósitos de S3 para aplicaciones internas, equipos o situaciones con varias cuentas.
- Genere objetos prefirmados URLs para permitir el acceso a corto plazo a los objetos de S3, adecuados para contenido descargable o cargas temporales, como registros de clientes.
- Usa Amazon CloudFront con cookies URLs o firmadas para ofrecer contenido privado de forma más segura a los usuarios autenticados

GAMESEC04-BP02 Limite el acceso de origen a las redes de entrega de contenido autorizadas () CDNs

Impida que los usuarios eludan sus redes de entrega de contenido para acceder directamente al contenido de su origen, como sus buckets de Amazon S3. Es importante restringir el acceso a su origen únicamente a las personas autorizadas CDNs, lo que reduce los costos de transferencia de datos que se derivan de la publicación innecesaria de contenido fuera del origen. También mejora tu nivel de seguridad, ya que permite que el acceso público al contenido de origen pase por el mismo punto de entrada, donde puedes implementar controles de seguridad periféricos, como el filtrado de AWS WAF nivel 7, la inyección e inspección de parámetros de solicitud HTTP relacionados con la seguridad y las protecciones distribuidas contra la denegación de servicio. DDo

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Para implementar estos controles para un origen de Amazon S3, puede usar una [identidad de acceso al CloudFront origen \(OAI\) de Amazon](#), que verifica que las solicitudes a sus objetos de S3 se originan en su CloudFront distribución. AWS WAF Asícielo a su CloudFront distribución para proporcionar un filtrado de capa 7. Sin embargo, si estás publicando contenido desde una plataforma adicional CDNs, puedes configurar la CDN para que inserte uno o más encabezados HTTP personalizados en las solicitudes de origen, que se pueden inspeccionar AWS WAF para comprobar que el tráfico entrante proviene de tu proveedor de CDN autorizado.

Este enfoque también es útil para evitar que los usuarios eludan a tus proveedores de CDN cuando tu origen está alojado en un [Application Load Balancer](#) (ALB). ALBs se puede asociar a él para obtener protecciones de capa 7. AWS WAF Puedes configurarlo AWS WAF para insertar un encabezado HTTP personalizado que será inspeccionado por tu ALB para procesar e inspeccionar el tráfico entrante al balanceador de cargas. AWS WAF

Ejemplo de cliente

AnyCompany Games implementa restricciones de acceso a Origin para proteger los recursos del juego, el contenido descargable y los archivos de parches del acceso directo no autorizado, lo que podría permitir a los jugadores eludir los controles de seguridad u obtener contenido premium sin la debida autenticación. Este enfoque les permite monitorear los patrones de acceso al contenido a través de un punto centralizado, lo que les permite identificar fácilmente los comportamientos de descarga sospechosos que podrían indicar la presencia de ataques coordinados o de una redistribución de contenido no autorizada.

Pasos para la implementación

- Utilice la identidad de acceso de CloudFront origen (OAI) de Amazon para restringir el acceso directo a los objetos S3
- AWS WAF Conéctese a CloudFront nuestro ALB para proporcionar un filtrado de capa 7 y protegerse contra los ataques DDoS y las solicitudes maliciosas.
- Configura encabezados HTTP personalizados en Cloudfront para comprobar que el tráfico entrante proviene de fuentes autorizadas.

GAMESEC04-BP03 Implemente restricciones geográficas para limitar el acceso no autorizado

Cuando un reproductor solicita tu contenido, Amazon entrega CloudFront el contenido solicitado desde la ubicación perimetral más cercana, independientemente de dónde se encuentre el reproductor. Sin embargo, puede haber situaciones en las que necesites restringir la forma en que los usuarios de determinadas partes del mundo pueden acceder a tu contenido. Por ejemplo, es posible que tengas una estrategia de despliegue de juegos continuos que publique contenido por fases o country-by-country que tengas que cumplir con los controles de acceso específicos de cada país.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Puedes usar [las restricciones geográficas](#), también conocidas como bloqueo geográfico, para impedir que los jugadores de ubicaciones geográficas específicas accedan al contenido que estás distribuyendo a través de una CloudFront distribución. Esta función te permite restringir el acceso a los archivos asociados a una distribución y restringir el acceso a nivel de país. Como alternativa,

puede utilizar un servicio de geolocalización de terceros para restringir el acceso a un subconjunto de los archivos asociados a una distribución o para restringir el acceso de forma más detallada que a nivel de país.

Al utilizar restricciones CloudFront geográficas, puedes permitir que tus jugadores solo accedan a tu contenido si se encuentran en uno de los países que figuran en una lista de países permitidos de países aprobados. También puedes impedir que tus jugadores accedan a tu contenido si se encuentran en uno de los países que figuran en una lista de países prohibidos. Si recibes una solicitud desde una ubicación geográfica bloqueada, CloudFront se devolverá al jugador un código de estado HTTP prohibido 403. Es importante tener en cuenta que esto funciona bien para contenido no confidencial y no debe usarse como protección independiente para la información de identificación personal o los artefactos confidenciales del juego.

Pasos para la implementación

- Usa restricciones CloudFront geográficas para permitir o denegar el acceso al contenido en función de las listas de permitidos o rechazados a nivel de país.
- Devuelve un código de estado HTTP prohibido 403 para las solicitudes que se originen en ubicaciones geográficas bloqueadas.
- Evite confiar únicamente en las restricciones geográficas para proteger el contenido confidencial o la información de identificación personal

GAMESEC04-BP04 Restrinja el acceso al contenido con soluciones de gestión de derechos digitales (DRM)

Considera la posibilidad de restringir el acceso al contenido de tus juegos mediante el uso de herramientas de cifrado sólidas, como una solución [de gestión de derechos digitales \(DRM\)](#). Este tipo de solución se puede utilizar para cifrar tu contenido privado y distribuir las claves de descifrado a los jugadores autorizados.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Las soluciones DRM se recomiendan en situaciones en las que quieres permitir a los jugadores descargar el contenido del juego antes de tiempo, pero no quieres que puedan acceder al contenido o reproducirlo hasta un tiempo predeterminado. Por ejemplo, esto es habitual en situaciones en las que los jugadores pueden reservar un juego y configurar su cliente de juego para que comience a

descargar automáticamente los archivos cifrados antes de tiempo. Esta estrategia verifica que el juego se haya descargado y esté listo para jugarse una vez que se haya lanzado oficialmente. Una vez lanzado el juego, el cliente del juego puede solicitar las claves de descifrado a la solución de administración de DRM para que pueda descifrar los archivos descargados anteriormente y empezar a jugar.

Los sistemas DRM también se utilizan para bloquear la redistribución y manipulación no autorizadas de juegos después de que un jugador autorizado los haya descargado e instalado. Los sistemas DRM requieren la integración con el origen para intercambiar las claves de cifrado y autorizar a los jugadores a recuperar la clave de descifrado. Los proveedores comerciales de DRM ofrecen una gama de soluciones con funciones y soporte para diferentes dispositivos.

Pasos para la implementación

- Utilice soluciones DRM para cifrar el contenido privado del juego y distribuir las claves de descifrado a los jugadores autorizados.
- Habilita la descarga previa de archivos cifrados para los juegos reservados y desbloquea el acceso con claves de descifrado en el momento del lanzamiento.
- Integre los sistemas DRM con el origen para gestionar las claves de cifrado y bloquear la redistribución o manipulación no autorizadas del contenido.

Detección

GAMESEC05: ¿Cómo monitorizas y analizas el comportamiento de uso de los jugadores en tu juego?

Supervisar y analizar el comportamiento de uso de los jugadores es fundamental para los estudios de videojuegos, ya que les permite detectar amenazas a la seguridad, trampas y otras formas de comportamiento abusivo que podrían comprometer la integridad del juego y la seguridad de los jugadores. Al rastrear patrones como tasas de progreso inusuales, transacciones anormales en el juego o comportamientos de comunicación sospechosos, podrás identificar posibles tramposos, cuentas fraudulentas o amenazas coordinadas antes de que afecten significativamente a la experiencia del jugador.

Prácticas recomendadas

- [GAMESEC05-BP01 Implementa una estrategia integral de recopilación de datos para monitorear el comportamiento de los jugadores](#)
- [GAMESEC05-BP02 Recopila, almacena y analiza los registros de uso de los jugadores para detectar comportamientos inapropiados](#)

GAMESEC05-BP01 Implementa una estrategia integral de recopilación de datos para monitorear el comportamiento de los jugadores

Para mantener una experiencia de jugador positiva, implementa una estrategia integral de recopilación y análisis de datos. La captura, el almacenamiento y el análisis de los datos relevantes proporcionan información sobre cómo los jugadores interactúan con las funciones del juego y entre sí. Este enfoque basado en los datos puede guiar la toma de decisiones, mejorar la participación y la retención de los jugadores, optimizar las estrategias de monetización y, en última instancia, mejorar la experiencia general del jugador.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Implemente sistemas de recopilación de datos para capturar y registrar las acciones relevantes de los jugadores, como las sesiones de juego, el progreso, los logros, las compras, las interacciones con los elementos del juego y las actividades sociales. Recopile datos del servidor, como la carga del servidor, el tráfico de la red y los registros de errores, para supervisar el rendimiento técnico e identificar posibles problemas. Recopila los comentarios de los jugadores a través de encuestas, foros, solicitudes de asistencia y canales de redes sociales para entender sus experiencias y preferencias.

Al almacenar los datos del juego, establece un almacén de datos centralizado o un lago de datos para almacenar y organizar los datos recopilados e implementa procesos de limpieza, transformación y agregación de datos a fin de prepararlos para un análisis eficiente.

Tras almacenar los datos, analízalos para obtener información valiosa, como la retención y la pérdida de jugadores, las estrategias de monetización y el uso de funciones mediante herramientas de visualización de datos.

Pasos para la implementación

- Capture y registre las acciones de los jugadores, las métricas del servidor y los comentarios para supervisar las interacciones y el rendimiento técnico.

- Utilice un almacén de datos centralizado, como Amazon Redshift o S3 Data Lake, para almacenar, limpiar, transformar y organizar los datos del juego para su análisis.
- Analice los datos recopilados con herramientas de visualización, como Amazon Quicksight, para obtener información sobre la retención de jugadores, la monetización y el uso de funciones.

GAMESEC05-BP02 Recopila, almacena y analiza los registros de uso de los jugadores para detectar comportamientos inapropiados

Instrumente su juego para recopilar registros y comprender cómo los jugadores usan las funciones de su juego y cómo interactúan con otros jugadores. A continuación, podrás bloquear las actividades no autorizadas que puedan degradar la experiencia de los jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

[Envía eventos de registro estructurados a Game Analytics Pipeline, mediante una solución de registro como Amazon CloudWatch Logs o Amazon OpenSearch Service, o mediante una solución de un AWS socio como Datadog, Sumo Logic, New Relic, Honeycomb.io o Splunk.](#) Estructura estos registros de uso de los jugadores para que puedan usarse para detectar cuándo es necesario investigar acciones específicas de los jugadores.

Una vez que hayas capturado los datos, considera la posibilidad de implementar herramientas para detectar un comportamiento de uso inapropiado. Por ejemplo, si tu juego tiene funciones sociales, como la mensajería de los jugadores, el chat de voz o los foros en línea, guarda los registros de las interacciones de estos jugadores en un formato que pueda analizarse con fines de moderación.

Configure la función de chat de voz del juego para exportar las grabaciones a Amazon S3 y utilice [Amazon Transcribe](#) para convertir la voz de audio a un formato de texto que se pueda almacenar para su procesamiento. Como alternativa, puedes realizar transcripciones de streaming en tiempo real integrando el servicio de chat de voz del backend del juego directamente con la API Transcribe [para transcribir el audio en streaming](#) en tiempo real. Los equipos de moderación pueden revisar el contenido manualmente y, una vez que esté en un formato estándar, también puedes usar los servicios de AWS inteligencia artificial y aprendizaje automático para realizar la moderación automáticamente. [Amazon Comprehend](#) se puede utilizar para procesar el lenguaje natural (PNL) a fin de descubrir información del texto no estructurado, lo que permite clasificar y organizar las conversaciones en temas relevantes e identificar comportamientos inapropiados, como blasfemias.

Pasos para la implementación

- Recopile, almacene y analice los registros de uso de los jugadores.
- Utiliza AWS los servicios de inteligencia artificial y aprendizaje automático para revisar y obtener información de forma más eficiente sobre los registros de uso de los jugadores.

Protección de la infraestructura

Consulte el documento técnico de Well-Architected Framework para conocer las mejores prácticas de protección de la [infraestructura para la seguridad que se aplican a las](#) cargas de trabajo de los juegos.

GAMESEC06: ¿Cómo se supervisan las amenazas a la infraestructura y se responde a ellas?

Supervisar las amenazas a la infraestructura y responder a ellas es fundamental para los estudios de videojuegos, ya que su infraestructura representa la columna vertebral que sustenta a millones de jugadores simultáneos, procesa las transacciones con dinero real y almacena datos valiosos sobre los jugadores y contenido de juegos patentado. Es esencial que los estudios de videojuegos implementen sistemas de monitoreo y procesos de respuesta a incidentes que puedan preservar la integridad tanto de las experiencias de los jugadores como de las operaciones comerciales.

Prácticas recomendadas

- [GAMESEC06-BP01 Utilice herramientas para detectar y responder a las amenazas a su infraestructura](#)
- [GAMESEC06-BP02 Utilice herramientas de inteligencia artificial y aprendizaje automático para automatizar aspectos de su estrategia de protección de infraestructuras](#)
- [GAMESEC06-BP03 Utilice la información de los registros a nivel del sistema para mejorar continuamente su estrategia de protección de infraestructuras](#)

GAMESEC06-BP01 Utilice herramientas para detectar y responder a las amenazas a su infraestructura

Para supervisar continuamente las actividades maliciosas y los comportamientos no autorizados en su AWS entorno, considere la posibilidad de utilizar [Amazon GuardDuty](#). GuardDuty identifica

las amenazas mediante la supervisión del comportamiento de las cuentas, la actividad de la red y los patrones de acceso a los datos en su entorno. Analiza los eventos en varias fuentes de datos, como los registros de CloudTrail eventos, los registros de flujo de Amazon VPC y los registros de DNS para detectar posibles amenazas. Al integrarse con Amazon CloudWatch Events y Lambda, las GuardDuty alertas se pueden reenviar automáticamente a los equipos de seguridad pertinentes para su posterior análisis.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

[AWS Security Hub CSPM](#) proporciona una visión completa del estado de su seguridad AWS y compara su entorno con los estándares y las mejores prácticas del sector de la seguridad. Security Hub CSPM recopila datos de seguridad de todos Cuentas de AWS los servicios y productos de socios externos compatibles y analiza sus tendencias de seguridad e identifica los problemas de seguridad de mayor prioridad. La GuardDuty [integración de Amazon con el Security Hub CSPM](#) le permite enviar los resultados desde GuardDuty el Security Hub CSPM. Luego, Security Hub CSPM puede incluir esos hallazgos en su análisis de su postura de seguridad.

Es habitual que los delincuentes empleen bots para apoderarse de las cuentas y hacer trampas en los juegos. El [control de bots de WAF](#) te ofrece visibilidad y control sobre el tráfico de bots habitual y generalizado, que puede consumir recursos excesivos, distorsionar las métricas, provocar tiempos de inactividad o realizar otras actividades no deseadas.

El ransomware es un código malicioso diseñado para obtener acceso no autorizado a sistemas y conjuntos de datos y cifrar esos datos para bloquear el acceso de actores legítimos. Cuando el ransomware ha bloqueado el acceso de los jugadores a sus sistemas y ha cifrado sus datos confidenciales, los ciberdelincuentes exigen un rescate antes de proporcionarles una clave de descifrado para desbloquear los datos. Un evento malintencionado puede cerrar por completo las organizaciones, lo que supone costes significativos y una pérdida de productividad empresarial. Consulte [Proteger su Nube de AWS entorno contra el ransomware](#) para conocer las mejores prácticas que puede aplicar para reforzar su capacidad de combatir el ransomware antes, durante y después de que se produzca un incidente.

Tu juego puede ofrecer a los jugadores la posibilidad de ponerse en contacto con los agentes de asistencia al jugador a través de un centro de llamadas, como [Amazon Connect](#), o de los bots de chat que utilizan Amazon Lex. Amazon Connect ofrece soporte para la [supervisión de conversaciones grabadas y en directo](#). Para analizar las interacciones entre los jugadores y los bots

de chat de asistencia a los jugadores creados con Amazon Lex, puede almacenar los [registros de conversaciones](#) de estas interacciones en Amazon CloudWatch Logs, que se pueden exportar a Amazon S3 y analizar como se ha descrito anteriormente.

Por último, realice pruebas de penetración como parte de su estrategia de protección de la infraestructura. Ya sea que realice estas evaluaciones internamente o a través de un AWS socio, respete [las políticas de AWS atención al cliente para las pruebas de penetración](#).

Pasos para la implementación

- Utilice Amazon GuardDuty para supervisar el comportamiento de las cuentas, la actividad de la red y los patrones de acceso a los datos en busca de amenazas, e intégrele con Security Hub CSPM para obtener una visión de seguridad unificada.
- Implemente el control de AWS WAF bots para ayudar a detectar y mitigar el tráfico de bots que puede dañar los recursos y las experiencias de los jugadores.
- Realiza pruebas de penetración con regularidad, siguiendo las políticas de AWS atención al cliente, para evaluar y reforzar tu postura en materia de seguridad.

GAMESEC06-BP02 Utilice herramientas de inteligencia artificial y aprendizaje automático para automatizar aspectos de su estrategia de protección de infraestructuras

[Amazon Lookout for Metrics](#) utiliza el aprendizaje automático para detectar y diagnosticar automáticamente las anomalías en los datos operativos y empresariales, y supervisa las métricas más importantes para sus empresas con mayor velocidad y precisión. El servicio también facilita el diagnóstico de la causa raíz de las anomalías, como una caída repentina de los ingresos, los inicios de sesión, las transacciones o la retención. No es necesario que los desarrolladores de juegos tengan experiencia en aprendizaje automático para configurarlos y pueden conectarse a fuentes de datos populares, como Amazon S3, Amazon CloudWatch, Amazon RDS y Amazon Redshift, así como a muchas aplicaciones SaaS. Por ejemplo, puede [integrar Amazon Lookout for Metrics con Game Analytics](#) Pipeline y otras fuentes de datos para empezar a analizar el comportamiento y detectar anomalías.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Como alternativa, puede optar por crear, entrenar y alojar un modelo de aprendizaje automático personalizado con [Amazon SageMaker AI](#) para abordar casos de uso como la moderación de contenido, la detección de toxicidad, la detección de trampas, la detección de fraudes y más.

Ejemplo de cliente

AnyCompany Games utiliza Amazon Lookout for Metrics para detectar automáticamente patrones inusuales en el rendimiento del servidor, los intentos de inicio de sesión de los jugadores o los volúmenes de transacciones que podrían indicar amenazas de actores malintencionados. Además, han utilizado Amazon SageMaker AI para desarrollar modelos de aprendizaje automático personalizados que analizan continuamente los patrones de tráfico de la red y el comportamiento de los jugadores para ayudar a identificar amenazas coordinadas, como las redes de bots que intentan explotar su economía virtual.

Este enfoque automatizado permite a su equipo de seguridad centrarse en investigar las amenazas reales y responder a ellas, en lugar de monitorizar manualmente miles de métricas y, al mismo tiempo, garantizar que los patrones de amenazas emergentes se detecten y aborden antes de que puedan afectar significativamente a la disponibilidad de los juegos o a la seguridad de los jugadores.

Pasos para la implementación

- Utilice Amazon Lookout for Metrics para detectar y diagnosticar automáticamente anomalías en datos empresariales y operativos clave
- Integre Amazon Lookout for Metrics con fuentes de datos como Game Analytics Pipeline o Amazon S3, CloudWatch o para monitorizar métricas como los ingresos, los inicios de sesión y la retención.
- Utilice Amazon SageMaker AI para crear, entrenar y alojar modelos de aprendizaje automático personalizados para casos de uso avanzados, como la detección de trampas, la prevención del fraude y la moderación de contenido.

GAMESEC06-BP03 Utilice la información de los registros a nivel del sistema para mejorar continuamente su estrategia de protección de infraestructuras

[Capture y almacene los registros a nivel del sistema de los servicios pertinentes, como los registros de acceso al servidor S3, los registros de CloudFront acceso y los registros de acceso a ALB.](#) Estos

registros se pueden almacenar en un compartimento de S3 en tu cuenta y son útiles para asociar la información de uso de los jugadores dentro del juego con la información del sistema, incluidos los detalles de conexión, como las direcciones IP, los encabezados de las solicitudes y la manipulación y el filtrado de las solicitudes relevantes que hayas configurado en el backend del juego. Puede enviar estos registros a las mismas soluciones de registro mencionadas anteriormente y [analizarlos mediante consultas SQL con Amazon Athena](#) sin necesidad de mover los registros fuera de Amazon S3.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

El [analizador de acceso para S3](#) es una función que supervisa las políticas de acceso a los buckets y se asegura de que las políticas solo proporcionen el acceso previsto a los recursos de Amazon S3. Access Analyzer para S3 evalúa sus políticas de acceso a los buckets y le permite detectar y corregir rápidamente los buckets con un posible acceso no intencionado.

Pasos para la implementación

- Utilice AWS los servicios de detección de amenazas y respuesta a incidentes para automatizar algunos aspectos de su estrategia de protección de infraestructuras.
- Obtenga información sobre la protección de su infraestructura mediante registros y AWS servicios a nivel de sistema para la inteligencia artificial y el aprendizaje automático.

Protección de datos

Al desarrollar y diseñar la arquitectura de tu juego, ten en cuenta qué tipo de datos recopila tu estudio y cómo has decidido enfocarte en protegerlos. Entre los temas que hay que explorar en relación con este aspecto de la seguridad se incluyen los siguientes:

- Cómo ha decidido identificar y clasificar sus datos
- ¿Cómo protege los datos en reposo
- ¿Cómo se protegen los datos en tránsito

No existen mejores prácticas de protección de datos específicas para Games Lens. [Consulte el documento técnico de Well-Architected Framework para conocer las mejores prácticas de protección de datos para la seguridad.](#)

Respuesta a incidentes

GAMESEC07: ¿Cómo definís y aplicáis las políticas para responder a la mala conducta y el comportamiento abusivo de los jugadores?

La mala conducta y los comportamientos abusivos de los jugadores pueden afectar significativamente a la experiencia de tus jugadores. El mal comportamiento de los jugadores puede alejar a los jugadores legítimos, lo que se traduce en una menor retención de jugadores, una reducción de los ingresos por compras dentro del juego y de reseñas negativas que pueden dañar la reputación y las ventas futuras de un juego.

Define políticas que promuevan acciones positivas entre tus jugadores y determina cómo vas a hacerlas cumplir.

Prácticas recomendadas

- [GAMESEC07-BP01 Implemente un plan de respuesta a incidentes para hacer frente a los malos actores y el comportamiento abusivo](#)
- [GAMESEC07-BP02 Prohíbe las cuentas que estén asociadas con actores maliciosos](#)

GAMESEC07-BP01 Implemente un plan de respuesta a incidentes para hacer frente a los malos actores y el comportamiento abusivo

Crea un plan de acción para responder a los malos actores y al comportamiento abusivo en tu juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Ten en cuenta factores como cuándo suspender temporalmente o prohibir permanentemente a los jugadores y durante cuánto tiempo debes deshabilitar las credenciales de los jugadores suspendidos temporalmente.

Ejemplo de cliente

AnyCompany Games crea un sistema escalonado de respuesta a incidentes en el que las infracciones menores, como los mensajes de chat inapropiados, dan lugar a la suspensión

automática de la cuenta durante 24 horas, mientras que las infracciones más graves, como las trampas o el acoso, conllevan suspensiones inmediatas de 7 días, cuya revisión es obligatoria por parte de moderadores humanos.

Además, AnyCompany Games establece procedimientos de escalamiento en los que los infractores reincidentes se enfrentan a suspensiones cada vez más largas. Crean procesos de apelación que permiten a los jugadores falsamente señalados impugnar las acciones automatizadas y, al mismo tiempo, mantienen la seguridad mediante requisitos de verificación de identidad.

GAMESEC07-BP02 Prohíbe las cuentas que estén asociadas con actores maliciosos

Si no se mitigan, los comportamientos abusivos en un juego pueden seguir teniendo un impacto negativo en la experiencia de juego de otras personas y deben mitigarse lo antes posible. Implementa un proceso que imponga prohibiciones u otros tipos de restricciones a quienes se confirme que infringen tus condiciones de servicio.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Por lo general, las normas y el proceso de evaluación para determinar las circunstancias que justifican la imposición de este tipo de restricciones los determinará personal, como un equipo de la comunidad de jugadores o un equipo de confianza y seguridad de la organización. Una vez que hayas identificado a los actores infractores, ejecuta un flujo de trabajo predeterminado para actuar en función de los actores identificados.

Por ejemplo, [AWS Step Functions](#) las [AWS Lambda](#) funciones se pueden usar para ejecutar un flujo de trabajo automatizado que acepte un lote de cuentas de jugadores como entrada. A continuación, el flujo de trabajo actualiza las entradas de una tabla de [Amazon DynamoDB](#) denominada Prohibiciones, que pueden incluir detalles sobre la cuenta del jugador, el motivo de la prohibición y la duración.

En función del diseño del juego y del sistema de administración de cuentas y del tipo de abuso al que se someta por parte de personas malintencionadas, mantenga un sistema de registro de prohibiciones independiente del sistema de administración de cuentas. Es posible que no desees desactivar la cuenta del jugador desde tu sistema de administración de cuentas, sino simplemente desactivar su capacidad de jugar a tu juego. Esto puede resultar útil en situaciones en las que las

credenciales de la cuenta del jugador se utilizan para acceder a varios juegos con condiciones de servicio o políticas diferentes.

Pasos para la implementación

- Defina y haga cumplir las políticas para responder a los comportamientos abusivos de los malos actores.
- Utilice AWS los servicios para automatizar sus respuestas a los malos actores.

Recursos

- [AWS Guía técnica de respuesta a incidentes de seguridad](#)
- [AWS Blog de Machine Learning: detecte usuarios reales y reales y disuada a los malos actores con Amazon Rekognition Face Liveness](#)
- [AWS Soluciones para juegos: Community Health](#)

Seguridad de las aplicaciones

GAMESEC08: ¿Cómo aseguras tu CI/CD cartera?

Un proceso de desarrollo CI/CD de juegos suele estar compuesto por servidores de control de código fuente y almacenamiento de alta disponibilidad, recursos informáticos para ejecutar las compilaciones y software para realizar pruebas automatizadas, además de la conectividad de red adecuada desde las máquinas de desarrollo. Proteger tu CI/CD cartera de distribución es importante para proteger la información confidencial, preservar la integridad del código y mantener versiones fiables. La integración de la gobernanza y las barreras de protección permite a los desarrolladores ser ágiles y, al mismo tiempo, mantener las buenas prácticas de seguridad.

Dado que los juegos suelen gestionar el procesamiento de pagos, almacenar información personal y mantener economías virtuales que valen dinero real, una violación de la seguridad en el proceso de desarrollo podría provocar importantes pérdidas financieras, sanciones reglamentarias y una pérdida de confianza entre los jugadores.

Al integrar las medidas de seguridad, las organizaciones mantienen la visibilidad y el control del proceso de entrega del software, lo que permite responder rápidamente a los incidentes y promover una cultura de prácticas de codificación seguras.

Prácticas recomendadas

- [GAMESEC08-BP01 Aplique la seguridad en todas las etapas de la canalización de CI/CD](#)

GAMESEC08-BP01 Aplique la seguridad en todas las etapas de la canalización de CI/CD

Las barreras, como los controles de acceso, la separación de funciones y los registros de auditoría, brindan protección contra el acceso no autorizado o las actividades maliciosas.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Su personal, sus procesos y su tecnología también deberían proteger la cartera. Las personas más cercanas al código deben establecer prácticas de codificación seguras y asegurarse de que las siguen. Repite tus procesos de forma continua para comprobar que el nivel de seguridad es uniforme en todo el proceso. Por último, implemente tecnología para verificar que no se pasen por alto las mejores prácticas y procesos.

Ejemplo de cliente

AnyCompany Games implementa controles de acceso basados en roles, en los que solo los desarrolladores experimentados pueden aprobar los cambios en el código de su sistema antitrampas, al tiempo que exigen que los miembros del equipo de seguridad revisen obligatoriamente el código de los componentes que gestionan los datos de pago de los jugadores.

Su programación CI/CD realiza automáticamente comprobaciones de validación de los modelos de amenazas para garantizar que las nuevas funciones, como un mercado de intercambio de jugadores, se prueben contra vectores de ataque previamente identificados, como la duplicación de artículos o los intentos de transacción fraudulenta.

Pasos para la implementación

- Proporcione a los usuarios permisos según el principio del privilegio mínimo.

- Se utiliza AWS CloudTrail para auditar las llamadas a la API realizadas en los servicios utilizados en la canalización.
- Utiliza enlaces previos a la confirmación para comprobar que el código sigue las prácticas generales y las políticas de la empresa.

Automatice la seguridad

GAMESEC09: ¿Cómo automatizas la seguridad en tu CI/CD cartera de proyectos?

Integre las medidas de seguridad en su CI/CD proceso para mantener una postura de seguridad sólida durante todo el ciclo de vida del desarrollo. Este proceso ofrece muchos de los mismos beneficios que tener la seguridad de su proceso. Tener una CI/CD cartera segura reduce la probabilidad de que se produzcan incidentes de seguridad que podrían retrasar el cronograma de desarrollo del juego.

Proporcionar seguridad en tu CI/CD proceso implica implementar las mejores prácticas y herramientas de seguridad en cada etapa del ciclo de desarrollo. Tener la seguridad en proceso también permite reducir el tiempo que se tarda en realizar las revisiones de seguridad.

Automatizar la seguridad de su CI/CD proceso es especialmente importante para comprobar que los controles de seguridad se implementan y prueban de forma coherente con cada cambio de código. La implementación de las herramientas y la automatización adecuadas puede ofrecer juegos seguros y protegidos.

Prácticas recomendadas

- [GAMESEC09-BP01 Integre las herramientas y la automatización para reducir el tiempo medio de las revisiones de seguridad](#)

GAMESEC09-BP01 Integre las herramientas y la automatización para reducir el tiempo medio de las revisiones de seguridad

Para identificar las vulnerabilidades de seguridad, las organizaciones pueden usar una variedad de herramientas y servicios diferentes, como las pruebas de seguridad de aplicaciones estáticas (SAST) y las pruebas dinámicas de seguridad de aplicaciones (DAST). El SAST es una forma de revisar el

código fuente y determinar la vulnerabilidad de seguridad. El DAST es una forma de caja negra de probar el código que pone a prueba las aplicaciones sin mirar el código fuente.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Otra herramienta que las organizaciones pueden utilizar es el análisis de composición del software (SCA), que evalúa la seguridad de las dependencias de terceros o de código abierto. Para un enfoque más manual, las revisiones seguras del código se pueden implementar a lo largo de todo el proceso.

Ejemplo de cliente

AnyCompany Games utiliza las herramientas SAST para detectar automáticamente los posibles fallos de seguridad durante el proceso de desarrollo. También utilizan las herramientas DAST para simular las amenazas contra las versiones de juegos en ejecución y comprobar que los controles de seguridad funcionan según lo previsto. Además, AnyCompany Games integra herramientas de análisis de dependencias en su proceso de desarrollo para identificar automáticamente las vulnerabilidades conocidas en bibliotecas y motores de juegos de terceros.

Pasos para la implementación

- Usa Amazon CodeGuru como herramienta de SAST.
- Utilice herramientas de código abierto como OWASP Dependency Check, o. SonarQube OWASPZap

Recursos

- [Seguridad para desarrolladores](#)

Modelado de amenazas

GAMESEC10: ¿Cómo se integra el modelado de amenazas en el ciclo de vida de desarrollo de aplicaciones de su organización?

El modelado de amenazas es el proceso de identificar y priorizar las posibles amenazas para su aplicación y determinar las soluciones que se pueden utilizar para mitigarlas. Esta práctica se ha vuelto cada vez más importante a medida que los juegos se han convertido en sistemas complejos y conectados que gestionan datos confidenciales de los usuarios y transacciones con dinero real.

Integre el modelado de amenazas como un ejercicio continuo para respaldar la seguridad del juego, no solo en la fase inicial de diseño, sino también a medida que el juego siga creciendo y evolucionando.

Prácticas recomendadas

- [GAMESEC10-BP01 Determine cuándo y cómo completar los ejercicios de modelado de amenazas a lo largo del ciclo de vida de desarrollo de sus aplicaciones](#)

GAMESEC10-BP01 Determine cuándo y cómo completar los ejercicios de modelado de amenazas a lo largo del ciclo de vida de desarrollo de sus aplicaciones

No existe una única forma mejor de abordar el modelado de amenazas. Los detalles sobre cuándo y cómo hacerlo variarán en función de las necesidades únicas de tu estudio de videojuegos. Por ejemplo, según el tamaño de tu estudio, es posible que tengas miembros del equipo que participen en uno o varios aspectos del proceso de modelado de amenazas.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

El [blog AWS de seguridad](#) proporciona una descripción general de las consideraciones que debe tener en cuenta a la hora de diseñar su estrategia de modelado de amenazas, tales como:

- ¿Cuáles miembros y personas de su equipo deberían participar en el modelado de amenazas
- ¿Cómo determinar las herramientas de flujo de trabajo adecuadas que se deben utilizar
- ¿Cómo determinar la propiedad de varios aspectos del modelado de amenazas
- Cómo identificar y evaluar los controles de seguridad que se utilizarán en el diseño de la carga de trabajo

Ejemplo de cliente

AnyCompany Los juegos comienzan por catalogar activos valiosos, como los datos de los jugadores, el código y los algoritmos del juego, las monedas del juego, el contenido generado por los usuarios y la propiedad intelectual, como el contenido inédito o los motores propietarios. Tienen en cuenta diferentes tipos de posibles actores perjudiciales, como los tramposos que buscan ventajas injustas, los delincuentes que intentan robar datos personales o financieros y los usuarios malintencionados que intentan interrumpir el juego.

Durante todo el proceso de desarrollo, AnyCompany Games utiliza modelos de amenazas para guiar las prácticas de codificación segura e influir en las estrategias de prueba para centrarse en las áreas de alto riesgo. Antes del lanzamiento de un juego, llevan a cabo exhaustivas revisiones de los modelos de amenazas para evaluar si están preparados para afrontar el aumento previsto de jugadores y los intentos de acceso no autorizado, y para preparar los procedimientos de respuesta a los incidentes.

Pasos para la implementación

- Implemente barreras en todas las etapas de su proceso. CI/CD
- Utilice la automatización y las herramientas para mejorar la eficiencia de las revisiones de seguridad de sus aplicaciones.
- Utilice el modelado de amenazas como un proceso para mejorar la seguridad de sus aplicaciones.

Recursos

- [AWS Blog de seguridad: ¿Cómo abordar el modelado de amenazas](#)
- [NIST: Guía para el modelado de amenazas de sistemas centrado en los datos](#)
- [El modelado de amenazas es la forma correcta para los constructores: formación virtual autodidacta de AWS Skill Builder](#)
- [Modelado de amenazas para constructores: taller AWS](#)

Recursos

Consulta los siguientes recursos para obtener más información sobre nuestras prácticas recomendadas relacionadas con la seguridad.

Documentos relacionados:

- [Escenarios comunes de Amazon Cognito](#)

- [Uso de firmados URLs](#)
- [Utilice los flujos de canal para eliminar las blasfemias y el contenido confidencial de los mensajes del SDK de Amazon Chime](#)
- [Seguridad en Amazon GameLift](#)
- [Entrega segura de contenido con Amazon CloudFront](#)
- [Guía de respuesta de seguridad](#)
- [AWS Mejores prácticas para DDo la resiliencia](#)
- [Proteja su Nube de AWS entorno contra el ransomware](#)

Soluciones de socios relacionados:

- [Datadog](#)
- [Sumo Logic](#)
- [Splunk](#)
- [Honeycomb.io](#)
- [New Relic](#)
- [AWS Marketplace - Soluciones DRM](#)

Materiales de formación relacionados:

- [Introducción a Amazon Cognito](#)
- [Formación autodidacta sobre seguridad](#)

Fiabilidad

El pilar de la confiabilidad incluye la capacidad de un sistema para recuperarse de las interrupciones en la infraestructura o el servicio, adquirir recursos informáticos de forma dinámica para satisfacer la demanda y mitigar las interrupciones, como los errores de configuración o los problemas transitorios de la red.

Áreas de interés

- [Principios de diseño](#)
- [Principios básicos](#)
- [Arquitectura de la carga de trabajo](#)
- [Administración de cambios](#)
- [Administración de errores](#)
- [Recursos](#)

Principios de diseño

Además de los principios de diseño del documento técnico AWS Well-Architected Framework, los siguientes son principios de diseño que pueden aumentar la confiabilidad en la nube para las cargas de trabajo de juegos:

- Establezca los objetivos de máxima concurrencia de jugadores y escalabilidad del sistema necesarios para cumplir con las proyecciones empresariales: antes de lanzar un juego y durante las operaciones de juego en directo, calcule el número de jugadores simultáneos que se espera que tengan en su punto álgido para establecer objetivos de escalabilidad del sistema a fin de cumplir con estas proyecciones. Esto ayuda a establecer una base para la fiabilidad del juego. Defina políticas de escalado que se adapten automáticamente a los cambios en la demanda sin que ello afecte a la disponibilidad, verificando que sus sistemas de escalado gestionen correctamente las sesiones de los jugadores activos.
- Mide tu fiabilidad y el impacto en la experiencia del jugador: define los indicadores clave de rendimiento (KPIs) que representen el estado de tu juego. Supervisa el impacto de los cambios en la infraestructura y las funciones del juego en tu fiabilidad.

Principios básicos

Para lograr la confiabilidad, un sistema debe contar con una base bien planificada y contar con mecanismos de monitoreo que permitan gestionar los cambios en la demanda o los requisitos. El sistema debe diseñarse para detectar las fallas y repararse automáticamente.

No hay una base de mejores prácticas específicas para los juegos. Consulte el documento técnico de Well-Architected Framework para conocer las mejores prácticas sobre los fundamentos de la confiabilidad que se aplican a las [cargas](#) de trabajo de los juegos.

Arquitectura de la carga de trabajo

GAMEREL01: ¿La arquitectura de su juego aprovecha la resiliencia de la nube?

AWS la infraestructura se basa en regiones y zonas de disponibilidad. Regiones de AWS proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que se conectan mediante redes de baja latencia, alto rendimiento y alta redundancia. Estas estructuras se pueden usar para diseñar cargas de trabajo centrándose en los objetivos de confiabilidad.

Prácticas recomendadas

- [GAMEREL01-BP01 Distribuya la infraestructura de juegos en múltiples zonas y regiones de disponibilidad para mejorar la resiliencia](#)

GAMEREL01-BP01 Distribuya la infraestructura de juegos en múltiples zonas y regiones de disponibilidad para mejorar la resiliencia

Para minimizar el impacto de las deficiencias de la infraestructura localizada en sus jugadores, debe distribuir el despliegue de la infraestructura de manera uniforme en ubicaciones independientes suficientes para poder soportar las deficiencias inesperadas y, al mismo tiempo, tener suficiente capacidad para satisfacer las necesidades de los jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

A la hora de implementar la infraestructura de juego, se recomienda distribuir la capacidad de manera uniforme entre las distintas zonas de disponibilidad de una región, de modo que puedas resistir las interrupciones en una o más zonas de disponibilidad sin que ello afecte a la experiencia del jugador. Los servicios de backend de juegos, como las aplicaciones web, deben tener una carga equilibrada en varias zonas de disponibilidad o deben crearse mediante un servicio gestionado, como AWS Lambda Amazon API Gateway, que proporciona una alta disponibilidad regional por diseño. Del mismo modo, los componentes que mantienen el estado, como las cachés, las bases de datos, las colas de mensajes y las soluciones de almacenamiento, deben diseñarse para proporcionar una persistencia duradera de los datos en varias zonas de disponibilidad, lo que se proporciona por diseño en servicios como Amazon S3, DynamoDB y Amazon SQS, y se puede configurar en otros servicios.

Al diseñar la arquitectura de alojamiento de sus servidores de juegos para garantizar la resiliencia, implemente sus flotas de servidores de juegos de manera uniforme en las zonas de disponibilidad dentro de una Región de AWS para maximizar su acceso a la capacidad informática disponible en la región y reducir el alcance del impacto de las deficiencias de las zonas de disponibilidad. Por ejemplo, puede configurar [Amazon EC2 Auto Scaling](#) para que utilice las zonas de disponibilidad. Si una EC2 instancia deja de estar en mal estado, EC2 Auto Scaling puede reemplazarla y lanzar instancias en otras zonas de disponibilidad si una o más de las zonas de disponibilidad dejan de estar disponibles.

Para la infraestructura crítica, como la autenticación, aprovisiona una cantidad mínima de instancias viables que se ejecuten en varias zonas de disponibilidad y utilice el escalado automático para gestionar los aumentos de carga o la tolerancia a errores si se produce un deterioro en una de las zonas de disponibilidad.

Implemente su infraestructura de juegos en varias regiones para maximizar la disponibilidad. Las funciones de recuperación ante desastres interregionales, como las bases de datos globales de Aurora y la infraestructura redundante que puede activarse con un simple cambio de DNS implementado en una región secundaria, pueden proporcionar continuidad del servicio en caso de que la región principal se vea afectada. Si bien te recomendamos que hagas esto para que los servicios de backend de tus juegos alcancen una alta disponibilidad, esta recomendación es especialmente importante para tus servidores de juegos.

Por ejemplo, en un juego multijugador, es probable que la capacidad de la infraestructura de los servidores de juegos supere las necesidades de capacidad de los demás servicios, ya que los servidores de juegos se utilizan para alojar las sesiones de juego de los jugadores. Muchos juegos

eligen dividir a los jugadores en regiones de juego lógicas (como el oeste y el este de EE. UU.). Para simplificar la experiencia del jugador y facilitar el uso de la infraestructura global para alojar juegos, considere la posibilidad de desvincular el nombre de las regiones de juego orientadas al jugador de la región del proveedor de la nube subyacente o de la ubicación del centro de datos que aloja físicamente los servidores del juego, junto con otras infraestructuras, como las Zonas Locales o sus propios centros de datos que alojan instancias de servidores de juegos compatibles con esa región de juego del jugador.

Al diseñar tu servicio de búsqueda de parejas, implementa una arquitectura multirregional con despliegues de software independientes en todas las regiones. Separa el despliegue del servicio de emparejamiento de las flotas que alojan las instancias de tus servidores de juegos para que puedas dirigir a los jugadores a un servidor de juego de las regiones, independientemente de la implementación regional de tu servicio de emparejamiento que haya gestionado la solicitud de emparejamiento.

Diseña una lógica en tu implementación de matchmaking para favorecer las regiones del servidor del juego que cumplan con tu latencia y otras reglas, con la posibilidad de redirigir a los jugadores a otras regiones si tus flotas tienen poca capacidad o si se producen otras interrupciones en la infraestructura regional.

Pasos para la implementación

- Distribuye la infraestructura del juego de manera uniforme en varias zonas de disponibilidad para ofrecer una alta disponibilidad y resiliencia.
- Implemente servicios de backend de juegos y componentes con estado mediante sistemas gestionados como Amazon S3 AWS Lambda, DynamoDB y SQS, o configure el equilibrio de carga y la durabilidad para soluciones personalizadas.
- Implemente despliegues en varias regiones para servicios y servidores de juegos críticos, utilizando soluciones de recuperación ante desastres como las bases de datos globales Aurora y las regiones lógicas orientadas a los jugadores desacopladas de las ubicaciones físicas subyacentes.

Recursos

- [Estabilidad estática](#)
- [Prácticas recomendadas para las colas de sesiones de GameLift juegos de Amazon](#)
- [Flotas GameLift multirregionales de Amazon](#)

- [Base de datos mundial Aurora](#) para recuperación ante desastres

Administración de cambios

GAMEREL02: ¿Cómo puedes escalar tus juegos con estado para adaptarlos a los cambios de la demanda?

Como la demanda de jugadores fluctúa con el tiempo, tu infraestructura de juegos debería poder ampliarse de forma adaptativa para hacer frente a estos requisitos cambiantes. Si bien es difícil predecir con antelación la popularidad de un juego, diseña un enfoque de arquitectura que permita añadir y eliminar capacidad de infraestructura para adaptarse a las fluctuaciones del número de jugadores.

Prácticas recomendadas

- [GAMEREL02-BP01 Implementa una estrategia de escalado que incorpore el estado de las sesiones de juego de los jugadores activos](#)
- [GAMEREL02-BP02 Support el uso de varios tipos de EC2 instancias para tu juego](#)

GAMEREL02-BP01 Implementa una estrategia de escalado que incorpore el estado de las sesiones de juego de los jugadores activos

Implemente una solución para escalar automáticamente la infraestructura de su juego de forma que incorpore la naturaleza funcional de las sesiones de los jugadores conectados activamente y gestione con elegancia las actividades de escalado sin interrumpir la jugabilidad.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Una de las ventajas de desarrollar un juego en la nube es la elasticidad que se puede lograr al escalar automáticamente la infraestructura del servidor según sea necesario para satisfacer la demanda. Si bien los juegos sin estado o asíncronos y los servicios de backend se pueden [escalar dinámicamente mediante las políticas de Amazon Auto Scaling, el escalado EC2 automático de EKS](#) o técnicas similares que se suelen adoptar para aplicaciones web escalables, los desarrolladores

de juegos suelen necesitar un enfoque más personalizado para escalar los juegos sincrónicos o con estado para ayudar a bloquear las interrupciones en las sesiones de los jugadores activos.

Pasos para la implementación

- En el caso de los juegos con estado, genera métricas personalizadas que se puedan usar para monitorear el estado de las sesiones de tus jugadores y la capacidad disponible del servidor de juegos, que se pueden informar a Amazon CloudWatch como métricas personalizadas. Funciones de ejercicio con monitoreo de aplicaciones, como CloudWatch Synthetics, que comprueban en el juego si hay alteraciones en las funciones que una simple monitorización de la salud hacia arriba y hacia abajo puede no detectar.
- Con métricas personalizadas, implemente un software de escalado de servidores de juegos, por ejemplo, como una aplicación sin servidor que utilice la AWS Lambda función o AWS Fargate para administrar la flota de instancias de servidores de juegos dedicadas mediante el AWS SDK para realizar llamadas a la API para actualizar la configuración de capacidad mínima, máxima y deseada para los [grupos de EC2 Auto Scaling](#) que alojan la versión de su servidor de juegos.
- Usa Amazon GameLift para alojar tus servidores de juegos y usa las [capacidades de escalado automático del servidor de out-of-the-box juegos](#) para administrar este proceso de escalado por ti.

Las capacidades de escalado automático de Amazon GameLift reconocen las sesiones de los jugadores activas y se pueden configurar para bloquear la finalización o la ampliación de las instancias de servidores de juegos que alojan activamente a los jugadores. Para obtener más información, consulte [Supervisar GameLift los servidores de Amazon con Amazon CloudWatch](#).

GAMEREL02-BP02 Support el uso de varios tipos de EC2 instancias para tu juego

Cuando alojes tu juego mediante EC2 instancias, o si utilizas contenedores alojados en EC2 instancias propias Cuenta de AWS, utiliza varios tipos de instancias en tu estrategia de alojamiento. Al usar varios tipos de instancias, puedes aumentar la cantidad de opciones de procesamiento que se pueden usar cuando el juego se está escalando para agregar más servidores y apoyar el crecimiento de jugadores, lo que mejora la confiabilidad en caso de que tu tipo de instancia preferido no esté disponible.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Cuando aloje su juego con instancias puntuales, utilice varios tipos de instancias, ya que la disponibilidad de las instancias puntuales fluctúa en función de la demanda de los clientes.

Pruebe su juego en varios tipos de instancias para cumplir con sus requisitos de costo y rendimiento y determine una clasificación priorizada de los tipos de instancias. Amazon EC2 Auto Scaling admite el uso de varios tipos y tamaños de instancias, así como la [asignación de pesos a cada tipo de instancia](#) de su configuración para que pueda implementar una clasificación priorizada de las opciones de procesamiento.

Cuando alojes tu juego con el alojamiento GameLift gestionado por Amazon, decide qué tipo de instancias necesita el juego y cómo ejecutar los procesos del servidor del juego en ellas (mediante una configuración de tiempo de ejecución). A la hora de elegir los recursos para una flota, ten en cuenta varios factores, como el sistema operativo del juego, el tipo de instancia (el hardware informático) y si vas a utilizar instancias bajo demanda, instancias puntuales o ambas. Los costes de alojamiento con Amazon dependen GameLift principalmente del tipo de instancias que utilices. Para obtener más información, consulte [Elegir recursos informáticos para una flota gestionada](#).

Pasos para la implementación

- Usa varios tipos de EC2 instancias para mejorar la confiabilidad y las opciones de escalado cuando alojes tu juego en contenedores EC2 o contenedores.
- Configure Amazon EC2 Auto Scaling o GameLift sus flotas con tipos y pesos de instancias priorizados para optimizar los costes y el rendimiento.
- Pruebe su juego en varios tipos de instancias para comprobar que el rendimiento cumple los requisitos y ajustar su estrategia de alojamiento en consecuencia.

Administración de errores

GAMEREL03: ¿Cómo se mantiene el estado del juego durante las interrupciones de la infraestructura?

Dado que la infraestructura del juego sufre diversos eventos operativos a lo largo del tiempo, la arquitectura del juego debe diseñarse para mantener la continuidad de las experiencias de los

jugadores y preservar el estado del juego durante los eventos de infraestructura. Para gestionar estos eventos, implementa mecanismos de supervisión, cierres estables y persistencia del estado para garantizar que tus jugadores disfruten de una experiencia de juego fluida.

Prácticas recomendadas

- [GAMEREL03-BP01 Supervise las interrupciones en los servidores de juegos y utilice los datos para mejorar la arquitectura de alojamiento y alcanzar los objetivos de confiabilidad](#)
- [GAMEREL03-BP02 Implementa una combinación flexible de funciones del juego para gestionar los fallos con un impacto mínimo en la experiencia del jugador.](#)
- [GAMEREL03-BP03 Supervisa los eventos de la infraestructura a lo largo del tiempo para medir el impacto en el comportamiento de los jugadores](#)

GAMEREL03-BP01 Supervise las interrupciones en los servidores de juegos y utilice los datos para mejorar la arquitectura de alojamiento y alcanzar los objetivos de confiabilidad

Supervisa las métricas de los servidores de juegos y el impacto de los fallos o las degradaciones en el rendimiento, como el aumento de la latencia bajo carga, en el comportamiento de los jugadores a lo largo del tiempo para que puedas ajustar tu estrategia de alojamiento de servidores de juegos a fin de cumplir con los requisitos de fiabilidad del juego. Si la infraestructura del servidor de juegos se deteriora, debe retirarse inmediatamente del servicio si está afectando a los jugadores o sustituirse de forma proactiva si no hay sesiones de jugadores activas alojadas en el servidor.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

En situaciones en las que los juegos se alojan en formato REST APIs, la fiabilidad del sistema se puede gestionar como en las arquitecturas de aplicaciones web tradicionales, en las que se puede equilibrar la carga del tráfico entre varios servidores de forma distribuida para mitigar el riesgo de que se produzcan fallos en los servidores.

Para una jugabilidad sincrónica en tiempo real, una sesión de juego suele estar alojada en un proceso de servidor de juegos que se ejecuta en una máquina virtual o en una instancia de servidor de juegos, ya que el estado del juego debe mantenerse de manera eficiente y replicarse en los clientes de juego conectados. Esta implementación implica que la experiencia del jugador está

estrechamente vinculada al rendimiento y la fiabilidad del proceso del servidor de juegos que aloja su sesión de juego. Este tipo de arquitectura hace que la gestión de la fiabilidad de los servidores de juegos sea más compleja que los enfoques tradicionales.

[Para mitigar el impacto de un fallo en el servidor del juego, configura el juego para que realice actualizaciones asíncronas continuas del estado del juego de un jugador en una caché o base de datos de alta disponibilidad, como Amazon ElastiCache \(Redis OSS\) o Amazon MemoryDB.](#) Si se produce un error en el servidor, el estado de la última partida guardada por el jugador se puede recuperar del almacén de datos externo y la sesión se puede restaurar en una nueva instancia de servidor de juegos.

Sin embargo, este enfoque añade costes y complejidad adicionales a la gestión de este estado externo y puede no ser adecuado para juegos de ritmo rápido o competitivos en los que los cambios de estado son tan frecuentes y se producen a una escala tan significativa que la introducción de un almacén de datos en caché en memoria de alto rendimiento provocaría un retraso de replicación demasiado significativo como para ser útil restaurar una sesión desde él. Para juegos de esta naturaleza, el enfoque óptimo es aceptar la pérdida del servidor y enviar al jugador de vuelta al lobby de un juego para buscar otra sesión o puedes redirigirlo automáticamente a otra sesión de juego.

Recopile la mayor cantidad de datos de registro útiles sobre la causa de la interrupción del servidor para poder investigar el problema más adelante. Amazon GameLift proporciona orientación para solucionar [problemas de flota y](#) ofrece la posibilidad de [acceder de forma remota a las instancias de la GameLift flota de Amazon.](#)

Pasos para la implementación

- Supervisa las métricas de los servidores de juegos para detectar posibles degradaciones en el rendimiento y elimina o reemplaza los servidores degradados según sea necesario para mantener la confiabilidad.
- Usa Amazon ElastiCache o MemoryDB para obtener actualizaciones asíncronas del estado del juego para permitir la recuperación de la sesión después de un fallo del servidor cuando sea posible.
- Capture datos de registro detallados sobre las interrupciones del servidor para su investigación y depuración, y aproveche herramientas como Amazon GameLift para la supervisión de la flota y el acceso remoto.

GAMEREL03-BP02 Implementa una combinación flexible de funciones del juego para gestionar los fallos con un impacto mínimo en la experiencia del jugador.

El desacoplamiento de los componentes se refiere al concepto de diseñar los componentes del servidor para que puedan funcionar de la forma más independiente posible. Algunos aspectos del juego son difíciles de disociar, ya que los datos deben estar lo más actualizados posible para ofrecer a los jugadores una buena experiencia de juego. Sin embargo, muchos componentes y tareas del juego se pueden disociar. Por ejemplo, las tablas de clasificación y los servicios de estadísticas no son fundamentales para la experiencia de juego, y las lecturas y escrituras de estos servicios se pueden realizar de forma asíncrona desde el juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Aplica una degradación correcta de las funciones de tu juego que puedan ser deshabilitadas automáticamente o por un administrador si se detectan problemas, y configura los servicios iniciales que dependen de la función para poder gestionar el error sin problemas. Por ejemplo, si los datos específicos de un jugador no se cargan correctamente en el cliente del juego, deberías considerar si estos datos son fundamentales para la experiencia de juego. Si no es así, configura el cliente del juego para que gestione correctamente este error sin interrumpir la experiencia del jugador, y opta por volver a intentar recuperar estos datos más adelante, cuando el jugador vuelva a visitar la pantalla.

Emplea lógicas como los tiempos de espera, los reintentos y los retrocesos para gestionar los errores y los fallos. Los tiempos de espera evitan que los sistemas se bloqueen durante períodos excesivamente largos. Los reintentos pueden proporcionar una alta disponibilidad de errores transitorios y aleatorios.

Defina los componentes no críticos que se puedan acoplar libremente a los componentes críticos. El acoplamiento flexible permite que los sistemas sean más resilientes, ya que los fallos en un componente no repercuten en otros. Si las funciones del juego no requieren conexiones activas a los servidores o al backend del juego, deberías implementar protocolos sin estado para escalar de forma dinámica y recuperarte de fallos transitorios. Desarrolle sus componentes no críticos de forma que puedan combinarse de forma flexible con protocolos sin estado mediante una API. HTTP/JSON Implemente las llamadas de red desde el cliente del juego para que sean asíncronas y no bloqueen,

a fin de minimizar el impacto que las funciones del juego u otros servicios dependientes pueden tener en los jugadores.

Para mejorar aún más la resiliencia mediante un acoplamiento flexible, utiliza un servicio de mensajería, como un sistema de colas, streaming o un sistema basado en temas entre componentes que se pueda gestionar de forma asíncrona. Este modelo es adecuado para una interacción que no requiere una respuesta inmediata o en la que basta con confirmar que se ha registrado una solicitud. Esta solución incluye un componente que genera eventos y otro que los consume. Los dos componentes no se integran mediante una point-to-point interacción directa, sino a través de un elemento intermedio, como un almacenamiento duradero o una capa de colas. Esto también ayuda a mejorar la confiabilidad del sistema al preservar los mensajes cuando se produce un error en el procesamiento.

Investigue y seleccione un mecanismo de mensajería adecuado, ya que los distintos servicios de mensajería tienen características diferentes, como los mecanismos de pedido y entrega. Diseñe las operaciones para que sean idempotentes, de modo que el sistema de mensajes elegido entregue los mensajes al menos una vez. A modo de ejemplo, consideremos un caso de uso típico de un juego en el que el juego necesita registrar el tiempo de juego de los jugadores, las estadísticas u otros datos relevantes, lo que puede llevar a un uso de alto rendimiento de escritura en momentos de máxima concurrencia de jugadores.

Para implementar una arquitectura fiable, considera si el caso de uso requiere read-after-write coherencia según la percepción del jugador. Por lo general, estos escenarios son adecuados para el procesamiento asíncrono y se pueden lograr mediante la implementación de un patrón de cola de escritura en el que las solicitudes se ingieren en una cola de mensajes escalable y duradera, como Amazon SQS, y se pueden insertar en la base de datos de backend en lotes mediante un servicio de consumo, como una función Lambda. Este enfoque es más fiable que la comunicación sincrónica entre varios componentes distribuidos, como el cliente del juego del jugador, los servidores web y de aplicaciones de backend y el sistema de base de datos interno. También reduce los costes, ya que no es necesario escalar la base de datos interna para alcanzar los picos de rendimiento de escritura, ya que el procesamiento realizado por los consumidores desde la cola de escritura puede utilizarse para reducir esta tasa de ingestión según sea necesario.

Pasos para la implementación

- Separa los componentes no esenciales, como las tablas de clasificación y los servicios de estadísticas, de las funciones fundamentales del juego para permitir operaciones asincrónicas y mejorar la resiliencia.

- Aplica una degradación adecuada de las funciones que no son esenciales con una lógica para los tiempos de espera, los reintentos y los retrasos, y comprueba que el cliente del juego gestiona los fallos sin interrumpir la experiencia del jugador.
- Utilice sistemas de mensajería como Amazon SQS para la comunicación asíncrona entre componentes, lo que permite un procesamiento escalable, duradero y fiable de casos de uso de alto rendimiento.

Recursos

- [Cree cargas de trabajo fiables y altamente escalables mediante una arquitectura de microservicios](#)
- [Integración de microservicios mediante el uso de servicios sin servidor AWS](#)
- [Descripción de la mensajería asíncrona para microservicios](#)
- [Introducción a los patrones de desarrollo de juegos escalables en AWS](#)
- Implementación de [Graceful Degradation](#)

GAMEREL03-BP03 Supervisa los eventos de la infraestructura a lo largo del tiempo para medir el impacto en el comportamiento de los jugadores

Supervisa el proceso del servidor de juegos, las métricas de las instancias del servidor de juegos y las métricas de la experiencia de juego para determinar la causa raíz de los problemas. Además de supervisar la CPU y la memoria, también puedes configurar la supervisión de las métricas de red relacionadas con las limitaciones de red de las EC2 instancias para avisarte de problemas como el exceso de ancho de banda u otros problemas a nivel de red que puedan indicar que los recursos del servidor están insuficientemente aprovisionados. packets-per-second

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Usa CloudWatch Synthetics para comprobar las funciones críticas de la aplicación para la experiencia del jugador, como la imposibilidad de iniciar sesión u otros problemas que afecten al servicio. En el caso de los servidores de juegos alojados en Amazon GameLift, considera monitorizar [métricas](#) como:

- GameServerInterruptions y InstanceInterruptions, lo que puede ayudar a comprender cómo las limitaciones en la disponibilidad de las instancias de Spot están afectando a los servidores de juegos que utilizan Spot.

- `ServerProcessAbnormalTerminations`, que se puede utilizar para detectar terminaciones anormales en los procesos del servidor de juegos.

Se recomienda mantener los datos estadísticos históricos sobre la fiabilidad del servidor de juegos. Utiliza estos datos históricos para elaborar informes y únelos a otros conjuntos de datos para descubrir posibles tendencias y evaluar el impacto en el comportamiento de los jugadores a lo largo del tiempo que puede deberse a problemas con el servidor del juego.

Amazon CloudWatch no conserva las métricas de forma indefinida y la [resolución de almacenamiento de las métricas](#) aumenta con el tiempo, así que considere la posibilidad de exportar estas métricas a un almacenamiento rentable a largo plazo, como Amazon S3. Puede configurar [CloudWatchMetric Streams](#) para que entregue automáticamente las métricas de [CloudWatchRegions](#) a su propio depósito de S3, donde se pueden almacenar a largo plazo en un nivel de almacenamiento como S3 Intelligent-Tiering y, finalmente, archivarlas con Amazon Glacier. Al colocar sus métricas en Amazon S3, estarán fácilmente disponibles para combinarlas con otros conjuntos de datos de su lago de datos para realizar consultas interactivas con Amazon [Athena](#).

Pasos para la implementación

- Supervise las métricas del servidor, la instancia y la red del juego, incluidos el ancho de banda y packet-per-second los límites, mediante Amazon CloudWatch y CloudWatch Synthetics para comprobar la funcionalidad de las rutas críticas.
- Realiza un seguimiento GameLift de métricas específicas, por `GameServerInterruption` ejemplo, `ServerProcessAbnormalTerminations` para evaluar el impacto de la disponibilidad de las instancias puntuales y detectar cierres anormales de servidores.
- Exporte CloudWatch métricas a Amazon S3 para un almacenamiento prolongado, utilice niveles rentables como S3 Intelligent-Tiering o Glacier y analice las tendencias con herramientas como Amazon Athena.

Recursos

- [Las métricas de rendimiento de la red EC2 a nivel de instancia de Amazon revelan nuevos conocimientos](#)
- [CloudWatchFlujos de métricas: envíe AWS métricas a sus socios y a sus aplicaciones en tiempo real](#)

Recursos

Consulta los siguientes recursos para obtener más información sobre nuestras prácticas recomendadas relacionadas con la fiabilidad.

Documentos relacionados:

- [Practicando la integración continua y la entrega continua en AWS](#)
- [Escalado automático de colas de trabajos asíncronos](#)
- [Diseño WorkloadService su arquitectura](#)
- [Se agotan los tiempos de espera, los reintentos y los retrasos por fluctuación](#)
- [Well-Architected Framework: pilar de confiabilidad](#)
- [Diseñando una arquitectura para una escalabilidad confiable](#)
- [La biblioteca de Amazon Builder](#)
- [Mensajería masiva en tiempo real para juegos multijugador](#)
- [Introducción a los patrones de desarrollo de juegos escalables en AWS](#)
- [Ejecutar microservicios en contenedores en AWS](#)
- [Alojamiento de aplicaciones web en la nube](#)
- [Creación de una infraestructura de red multiVPC escalable y segura](#)

Videos relacionados:

- [Re:Invent 2020: Ubisoft: creación de un juego multijugador multiplataforma en AWS](#)
- [re:Invent 2018: Supercell: Scaling Mobile Games](#)
- [re:Invent 2019: Cómo CAPCOM crea juegos divertidos con contenedores, datos y aprendizaje automático](#)
- [re:Invent 2018: globalizar las cuentas de los jugadores de Riot Games y mantener la disponibilidad](#)
- [re:Invent 2020: GameLoft - Un análisis profundo sobre la migración de lagos de datos sin tiempo de inactividad](#)

Formación relacionada:

- [Uso de Amazon GameLift Fleet IQ para servidores de juegos](#)
- [Alojamiento de servidores de juegos con Amazon EC2](#)

Eficiencia del rendimiento

El pilar de la eficiencia del rendimiento se centra en el uso eficiente de los recursos informáticos para cumplir con los requisitos y en mantener esa eficiencia a medida que la demanda cambia y las tecnologías evolucionan.

Adopte un enfoque basado en datos para seleccionar una arquitectura de alto rendimiento. Recopile datos exhaustivos sobre la arquitectura, desde el diseño de alto nivel hasta la selección y configuración de los tipos de recursos. Considere sus opciones arquitectónicas de forma cíclica para aprovechar el conjunto de servicios y soluciones en constante evolución. Las métricas ayudan a comprender las desviaciones respecto al rendimiento esperado para que pueda tomar medidas. Un enfoque basado en datos le ayuda a hacer concesiones con su arquitectura para mejorar el rendimiento, reducir los costes o mejorar la experiencia del desarrollador.

Áreas de interés

- [Principios de diseño](#)
- [Selección de la arquitectura](#)
- [Selección de región](#)
- [Desarrollo iterativo](#)
- [Computación y hardware](#)
- [Selección de cómputo](#)
- [Administración de datos](#)
- [Redes y entrega de contenido](#)
- [Proceso y cultura](#)
- [Recursos](#)

Principios de diseño

Además de los principios de diseño del documento técnico AWS Well-Architected Framework, los siguientes principios de diseño pueden lograr la eficiencia del rendimiento de sus juegos:

- **Mide el rendimiento de los juegos desde end-to-end:** Es importante medir el rendimiento tal y como lo perciben los jugadores. Esto significa que debes medir el rendimiento del cliente del juego, la infraestructura del juego y la conectividad a Internet que conecta a los jugadores

con la infraestructura. Esto te ayudará a entender dónde puedes mejorar el rendimiento de tu arquitectura.

- Optimice su arquitectura para mejorar las métricas que reflejen la experiencia real del jugador: a medida que vaya adaptando y evolucionando su arquitectura con el paso del tiempo, piense en cómo afectarán estas mejoras y cambios a la experiencia del jugador. Las cargas de trabajo de los juegos deberían poder resistir y minimizar el impacto de los fallos para bloquear las interrupciones generalizadas del juego. Las funciones y los sistemas del juego que no dependan de manera crítica unos de otros deberían estar desacoplados para reducir el radio de impacto de los fallos y aislar los problemas que afectan a los jugadores.
- Usa tecnologías que simplifiquen las operaciones del juego y aumenten la velocidad de desarrollo: da prioridad a la adopción de tecnologías que puedan mejorar la eficiencia de los desarrolladores. La sobrecarga operativa durante las fases de desarrollo previas a la producción puede distraer la atención de la mejora de la jugabilidad. Al aprovechar los servicios gestionados de AWS AWS nuestros socios, se puede reducir la ingeniería, lo que permite a los desarrolladores de juegos centrarse en el ciclo básico del juego y en la experiencia del jugador. Los requisitos de arquitectura y rendimiento pueden cambiar y evolucionar a lo largo del ciclo de vida del desarrollo del juego, por lo que se deben tener en cuenta las ventajas tecnológicas en cada fase.
- Diseñe la infraestructura para cumplir con los picos de concurrencia de jugadores y escalarla dinámicamente según sea necesario: la infraestructura debe diseñarse para adaptarse a la demanda de los jugadores. Las métricas, como la simultaneidad de las sesiones de los jugadores y el número de inicios de sesión, se pueden utilizar para escalar de forma preventiva antes de que los sistemas se sobrecarguen. Las métricas de uso reactivo del sistema, como el consumo de CPU y memoria, se pueden utilizar para escalar después de que los sistemas se sobrecarguen. Al escalar tu infraestructura de forma dinámica, puedes reducir los costes de funcionamiento del juego.

Selección de la arquitectura

GAMEPERF01: ¿Cómo seleccionas la opción de alojamiento adecuada para tus servidores de juegos?

La selección de la opción de alojamiento adecuada para sus servidores de juegos es fundamental para el rendimiento de los servidores de juegos. La decisión de utilizar EC2 instancias, una solución contenedora o un servicio totalmente gestionado es una de las primeras decisiones que se deben

tomar a la hora de diseñar una arquitectura para la producción. Cada opción de alojamiento tendrá diferentes capacidades y consideraciones a la hora de ajustar el rendimiento, escalar, operar e integrar.

Prácticas recomendadas

- [GAMEPERF01-BP01 Evalúe los requisitos de recursos y las necesidades de escalabilidad del servidor de juegos](#)
- [GAMEPERF01-BP02 Tenga en cuenta la sobrecarga operativa para escalar los servidores de juegos](#)
- [GAMEPERF01-BP03 Evalúe la integración con otros AWS servicios, entornos de desarrollo, arquitecturas de CPU objetivo y funciones](#)

GAMEPERF01-BP01 Evalúe los requisitos de recursos y las necesidades de escalabilidad del servidor de juegos

Evalúe los requisitos del servidor en función de sus necesidades de escalabilidad para comprobar que está seleccionando una opción de alojamiento que cumpla con sus requisitos y ofrezca un rendimiento óptimo.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Al seleccionar la opción de alojamiento adecuada para tus servidores de juegos, ten en cuenta los siguientes factores:

Requisitos de recursos del servidor de juegos

Evalúe los requisitos de CPU, memoria, red y almacenamiento de los procesos del servidor de juegos para determinar lo que consume el juego. No pases por alto las redes; cada fotograma requiere ciclos de CPU para recibir las acciones del jugador, actualizar el estado del juego y devolvérselo al jugador. Al descargar el procesamiento de paquetes, se puede liberar la CPU para utilizarla en las funciones principales del juego. La conexión en red es la base de un juego fluido y con capacidad de respuesta, por lo que probarlo al principio del proceso define un perfil de rendimiento básico para un juego.

Un juego de disparos en primera persona puede tener muchas acciones por segundo, lo que la CPU necesita para pasar rápidamente a la red, lo que puede favorecer las instancias de la familia C

optimizadas para la computación, mientras que un juego de estrategia por turnos que puede dedicar más ciclos de CPU a procesar por turno puede necesitar más memoria de las instancias de la familia R para almacenar y actualizar localmente el estado del juego en el servidor antes de devolverlo a los jugadores. Utilice un enfoque basado en datos, como [el método USE \(Utilization Saturation and Errors\)](#), para tomar decisiones arquitectónicas bien informadas.

Escalabilidad y elasticidad

Evalúe con qué rapidez y fluidez puede ampliarse cada opción de alojamiento para satisfacer la demanda de los jugadores sin comprometer el rendimiento. Ten en cuenta el nivel de automatización y flexibilidad necesarios para la carga de trabajo de tu juego a fin de mantener una experiencia de juego fluida durante las horas punta. Un servidor de juegos puede ampliarse rápidamente aumentando su utilización mediante la adición de procesos de servidor de juegos adicionales en la misma instancia, mientras que el backend de un juego puede escalar más lentamente en función del aumento del número de usuarios activos y de los juegos que se estén jugando. Tu flota debería ampliarse en función de la demanda para minimizar los costes y, al mismo tiempo, reducir al mínimo el tiempo de espera para que los jugadores accedan al juego. Consulte Amazon EC2 Spot Instance Advisor para obtener información sobre la rentable capacidad disponible para las flotas de servidores de juegos.

Pasos para la implementación

- Evalúe los requisitos de recursos de CPU, memoria, red y almacenamiento del servidor de juegos para seleccionar los tipos de instancias adecuados, teniendo en cuenta las necesidades de rendimiento específicas del juego, como el alto rendimiento de la red para los juegos de FPS o la optimización de la memoria para los juegos de estrategia por turnos.
- Compare diferentes opciones de alojamiento, como contenedores, instancias, servicios básicos y gestionados, analizando los datos de rendimiento mediante marcos como el método USE. Utilice estos conocimientos para tomar mejores decisiones sobre la arquitectura de su sistema.
- Diseñe flotas para que sean escalables y elásticas, aprovechando herramientas como EC2 Spot Instance Advisor para optimizar los costes y, al mismo tiempo, facilitar el escalado rápido para satisfacer la demanda de los jugadores durante las horas punta.

GAMEPERF01-BP02 Tenga en cuenta la sobrecarga operativa para escalar los servidores de juegos

Tenga en cuenta los gastos operativos y de administración asociados a cada opción de alojamiento.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Gastos generales operativos

Las soluciones autohospedadas en EC2 nuestros contenedores pueden proporcionar más control, pero también requerirán más administración. Los orquestadores de contenedores, como ECS o EKS, pueden reducir los tiempos de lanzamiento de los servidores en contenedores y, al mismo tiempo, aumentar la complejidad de las redes y la sobrecarga de organización del mantenimiento.

Por ejemplo, [los grupos de nodos gestionados por EKS](#) pueden automatizar el aprovisionamiento y la gestión del ciclo de vida de los servidores de juegos, pero no respetan los presupuestos de interrupción de los módulos al cerrar un nodo. Si el juego requiere un período de finalización superior a los 15 minutos para completar los juegos de forma segura, puede que tengas que crear enlaces de ciclo de vida o considerar la posibilidad de utilizar nodos autogestionados con controladores personalizados para bloquear la interrupción del juego.

Los servicios gestionados, como Amazon Game Lift, pueden gestionar la mayor parte de la sobrecarga operativa, pero reducen la cantidad de visibilidad y control sobre los requisitos especiales de la configuración de redes y seguridad de bajo nivel. La elección de una solución de servidor de juegos supone un equilibrio entre el nivel de personalización, control y responsabilidad que tendrás a la hora de ajustar el rendimiento del servidor de juegos y el comportamiento de escalado.

Pasos para la implementación

- Evalúe la sobrecarga operativa de las opciones de alojamiento y equilibre el esfuerzo de control y administración entre las soluciones autohospedadas EC2, como ECS o EKS, y los servicios gestionados, como Amazon Game Lift.
- Utiliza grupos de nodos gestionados por EKS para la automatización, pero implementa enlaces de ciclo de vida o controladores personalizados si tus servidores de juegos requieren períodos de finalización más largos que los predeterminados.
- Evalúe las desventajas entre la personalización, la visibilidad y la responsabilidad operativa a la hora de seleccionar una solución de servidor de juegos.

GAMEPERF01-BP03 Evalúe la integración con otros AWS servicios, entornos de desarrollo, arquitecturas de CPU objetivo y funciones

Evalúa qué tan bien se integra cada opción de alojamiento con otros AWS servicios de los que depende tu juego, como bases de datos, análisis o servicios de entrega de contenido.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Integración con otros servicios de AWS

La perfecta integración entre los servicios proporciona beneficios operativos, como una mejor supervisión del rendimiento y una entrega de datos segura y eficiente entre los componentes del juego, los servidores de juegos, los servicios de backend de los juegos y las soluciones de observabilidad.

Por ejemplo, coordinar los cambios de tráfico para los partidos en directo puede resultar complejo. Amazon Route 53 le ayudará a mantener sus registros de DNS actualizados, lo que simplifica los cambios de tráfico coordinados. AWS Los diales de tráfico de Global Accelerator le permiten enviar un porcentaje del tráfico a otra región y mantener el juego en funcionamiento durante el mantenimiento.

Entorno y herramientas de desarrollo

Tenga en cuenta las herramientas, marcos y entornos de desarrollo compatibles con cada opción de arquitectura. Comprueba que la opción elegida se ajuste a la solución de desarrollo de juegos y a los lenguajes de programación, ya que esto puede afectar a la capacidad de tu equipo de optimizar y mantener el rendimiento de los servidores de juegos. Ofrecer un juego en dispositivos móviles, consolas y PC aumentará la complejidad de las herramientas y las pruebas. El soporte multisistema es particularmente importante para los estudios de juegos múltiples, donde los servicios centralizados pueden estandarizar las mejores prácticas de desarrollo en todos los títulos.

La arquitectura y las características de la CPU de Target

Tenga en cuenta el perfil de rendimiento de los procesos del motor y del servidor de juegos y el nivel de compatibilidad con ARM disponible. Evalúe si puede beneficiarse de la mejora de la relación precio-rendimiento de los procesadores Graviton o x86 basados AMD64 en ARM. ¿Necesita utilizar funciones de Intel como el cifrado AES-NI, AVX o Turbo Boost? Revise los [tipos de hosts dedicados](#)

para identificar las familias de instancias de uno o varios sockets. Cuando utilices una familia de instancias con varios sockets, considera la posibilidad de fijar NUMA y compartir la caché L3 en los procesos del servidor de juegos. Usa la configuración de [estado C y P](#) para obtener el mejor rendimiento de tu juego ajustando el reloj de frecuencia y reduciendo los niveles de reposo.

Pasos para la implementación

- Seleccione opciones de alojamiento que se integren perfectamente con AWS servicios como AWS Secrets Manager ACM y otros para ayudar a agilizar la supervisión del rendimiento, proteger la entrega de datos y reducir las tareas operativas manuales.
- Compruebe la compatibilidad entre su opción de alojamiento y su entorno de desarrollo, marcos y lenguajes de programación para optimizar y mantener el rendimiento del servidor de forma eficaz.
- Evalúe los requisitos de arquitectura de la CPU, utilice Graviton para obtener una buena relación precio-rendimiento o x86 para funciones específicas, como AES-NI, AVX y Turbo Boost, y optimice el rendimiento del servidor con la fijación NUMA y el ajuste de estado C y estado P.

Selección de región

GAMEPERF02: ¿Cómo determinas en qué regiones geográficas alojarás tu infraestructura de juego?

Elegir la ubicación ideal para la infraestructura de tu juego puede mejorar el rendimiento de la red para los jugadores y el backend. Tener en cuenta desde dónde se conecta tu base de jugadores y cómo están construidas tus comunidades o servidores es importante para el crecimiento y la sostenibilidad a largo plazo de una región geográfica. La implementación de una infraestructura de servidores de juegos y servicios de back-end disociados puede beneficiar tu eficiencia operativa general y mejorar la resiliencia al utilizar múltiples regiones, zonas locales y centros de avanzada para alojar tu juego.

Prácticas recomendadas

- [GAMEPERF02-BP01 Selecciona una región local que esté cerca de tus jugadores](#)
- [GAMEPERF02-BP02 Diseñe un enfoque que permita colocar la infraestructura de juegos sensible a la latencia cerca de los jugadores para mejorar el rendimiento](#)

GAMEPERF02-BP01 Selecciona una región local que esté cerca de tus jugadores

Para el lanzamiento inicial de un juego, debes determinar dónde desplegar la infraestructura basándote en las conversaciones con las partes interesadas de la empresa, como los equipos de publicación, que determinan dónde se espera que el juego esté disponible para los jugadores y dónde centrarán sus esfuerzos de marketing y publicidad previos al lanzamiento.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Las partes interesadas de la empresa también deberían disponer de mecanismos para estimular la demanda a fin de comprender mejor la acogida y la viabilidad de los jugadores. Por ejemplo, estos equipos dispondrán de mecanismos como la reserva de juegos, campañas y eventos de marketing, listas públicas de correo electrónico para que los jugadores registren su interés antes del lanzamiento y otros métodos para establecer señales relevantes que permitan determinar dónde es probable que el juego tenga más jugadores en el momento del lanzamiento. El juego también puede utilizar una estrategia de lanzamiento regional que incluya fases de prueba y de lanzamiento temporal para determinar la demanda de jugadores regionales.

[Selecciona una región local](#) que esté cerca de tu base de jugadores y de tus desarrolladores y que cuente con los AWS servicios y funciones que necesitas para alojar tu juego. El hogar RRegion será el lugar donde se ejecutarán los servicios de back-end del juego, y es posible que también se ejecuten servidores de juegos. Evalúe una región de origen en función de los servicios compatibles, la conectividad con las ubicaciones periféricas, la proximidad a las regiones de conmutación por error y el número de zonas de disponibilidad. Si utiliza una zona local, tenga en cuenta que la región principal a veces se encuentra en un área geográfica diferente. A modo de ejemplo: la zona local us-east-1-scl-1a de Santiago de Chile tiene como región matriz la us-east-1 del norte de Virginia, a pesar de que está geográficamente más cerca de Sao Paulo sa-east-1.

Pasos para la implementación

- Identifica las regiones de despliegue en función de las señales de demanda de jugadores derivadas de las actividades previas al lanzamiento, como los pedidos anticipados, las campañas de marketing y el registro de intereses.
- Elige una región local cercana a la base de jugadores principal y a los desarrolladores, y asegúrate de que sea compatible con los AWS servicios necesarios, las ubicaciones periféricas y las regiones de conmutación por error.

- Evalúe las zonas locales cuidadosamente, teniendo en cuenta que la región principal puede diferir geográficamente de la ubicación de la zona local.

GAMEPERF02-BP02 Diseñe un enfoque que permita colocar la infraestructura de juegos sensible a la latencia cerca de los jugadores para mejorar el rendimiento

La ubicación separada de las infraestructuras sensibles a la latencia, como los servidores de juegos, minimiza el impacto de las rutas de red largas. Las implementaciones repetibles pueden facilitar el mantenimiento de varias ubicaciones que ofrezcan un mejor rendimiento a tus jugadores. El ping es una métrica común que aparece en la interfaz de usuario del juego, y un ping bajo puede ser una característica diferenciadora.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Al iniciar un juego por primera vez, es posible que aún no dispongas de suficiente información sobre tu base de jugadores para saber cuál es la mejor manera de implementar la infraestructura más cercana a los jugadores que están más interesados en jugar a tu juego. Se trata de un desafío habitual, por lo que debes prepararte para ello diseñando una arquitectura que te permita ajustar rápidamente tu estrategia de ubicación del alojamiento a fin de implementar los servidores donde los necesiten, más cerca de los jugadores. Lo habitual es que los desarrolladores de juegos evalúen periódicamente el despliegue de su infraestructura de juegos como un análisis periódico posterior al lanzamiento para invertir gradualmente en mejoras a lo largo del tiempo y con un enfoque iterativo.

Lo mejor es utilizar *infrastructure-as-code* plantillas, como AWS CloudFormation Terraform de Hashicorp, para la configuración de tu infraestructura, por ejemplo VPCs, las configuraciones de subred y las dependencias necesarias para lanzar servicios de juego fundamentales, de modo que puedas consultar estas plantillas, personalizarlas rápidamente si es necesario e implementarlas en ubicaciones donde se necesite infraestructura adicional para apoyar a tus jugadores.

También debe asegurarse de comprender cómo podría evolucionar su estrategia de implementación actual para permitir una expansión futura. Las plantillas de IaC se pueden repetir, pero no sustituyen a la planificación de la red. [IPAM gestiona](#) su VPCs El tamaño de las subredes, la selección de la zona de disponibilidad y el inventario de IP y la alineación de las zonas de disponibilidad entre cuentas. Es importante tener en cuenta la red, ya que puede ser perjudicial para los jugadores si se cambian. Los servidores de juegos desplegados en varias ubicaciones geográficas se conectarán al

backend del juego, que es más común si se alojan en una o varias regiones de origen, lo que puede requerir una configuración adicional para admitir la conectividad privada. Estas consideraciones deben evaluarse continuamente a lo largo del tiempo para que puedas realizar cambios en tu estrategia de alojamiento de juegos a medida que cambien los requisitos del juego o cambien los requisitos de los jugadores.

A la hora de determinar cuántas ubicaciones de alojamiento de juegos usarás para tu juego, ten en cuenta los siguientes factores:

- Mejora de la calidad de la experiencia del jugador: ¿En qué medida podrías mejorar la experiencia del jugador añadiendo ubicaciones de alojamiento de juegos adicionales? ¿Cuál es el aumento gradual de rendimiento que puedes lograr al hacerlo? ¿Cómo medirá esta mejora en el rendimiento?
- Qué grupos de jugadores debes priorizar: ¿Para cuántos jugadores puedes mejorar la experiencia si añades más ubicaciones de alojamiento de juegos? ¿A qué poblaciones de jugadores o ubicaciones geográficas darás prioridad?
- Impactos posteriores del cambio: si cambias tu estrategia de alojamiento de juegos, ¿cómo influirá esto en los tiempos de espera de los jugadores en las partidas? ¿El tamaño de las partidas, el equilibrio de habilidades o el número de jugadores del grupo de jugadores pueden adaptarse a un cambio de estrategia en la ubicación de alojamiento del juego? Si se admiten más ubicaciones, se puede fragmentar el grupo de jugadores y aumentar los costes y la complejidad.

Deberás evaluar cada una de estas consideraciones a la hora de determinar dónde añadir o eliminar ubicaciones de alojamiento de juegos. Por ejemplo, puedes optar por priorizar la mejora de la experiencia para los jugadores de las ubicaciones geográficas con menos experiencia de juego o para los jugadores que expresan los comentarios del público con mayor intensidad. También puedes optar por incluir la monetización de los jugadores entre tus prioridades, por ejemplo, centrando tu atención en mejorar la experiencia de los jugadores en ubicaciones geográficas que generen una fuente importante de ingresos para tu juego o que puedan generar ingresos incrementales si introduces mejoras en el rendimiento.

Además de alojar la infraestructura Regiones de AWS, puedes usar [las Zonas Locales](#), que son una extensión de una Región de AWS, para alojar tus servidores de juegos y otras aplicaciones sensibles a la latencia, como los servidores de chat de voz, más cerca de tus jugadores. También puedes optar por ejecutar la infraestructura de desarrollo de juegos en las Zonas Locales para mejorar la experiencia de tus equipos de desarrollo de juegos. Por ejemplo, puede utilizar las Zonas Locales para abordar casos de uso, como alojar réplicas de sus servidores de control de código fuente

autogestionados más cerca de los desarrolladores de juegos y ofrecer estaciones de trabajo virtuales de desarrollo de juegos y almacenamiento de contenido a los usuarios que utilizan EC2 instancias de Amazon, volúmenes de EBS y sistemas de FSx archivos de Amazon implementados en una o más zonas locales cercanas a sus estudios de desarrollo sin necesidad de alojar la infraestructura en las instalaciones.

Los [Outposts](#) son una buena opción cuando las Regiones o Zonas Locales no están disponibles en la misma área geográfica. AWS Debe tenerse en cuenta la conectividad de su centro de datos para que el servidor del juego pueda garantizar la fiabilidad del sistema de back-end. AWS Outposts y los servidores Outpost están diseñados específicamente para funcionar AWS en tu centro de datos con los mismos servicios y APIs para ayudar a crear un modelo de implementación coherente dondequiera que ejecutes el juego. Se pueden combinar varios racks en un Outpost lógico y la infraestructura se puede compartir entre ellos. Cuentas de AWS El ciclo de vida del hardware se gestiona mediante, AWS y el plazo de entrega puede ser de tan solo 3 meses.

Si está creando juegos con contenedores y desea tener la flexibilidad de adoptar una arquitectura de implementación híbrida con software de código abierto que se pueda implementar en su propia infraestructura local, puede usar [ECS Anywhere](#) o [EKS Anywhere](#) como alternativa a AWS Outposts nuestras Zonas Locales. Si hospedas en Amazon GameLift, [puedes usar Amazon GameLift Anywhere para ejecutar tu servidor basado en hardware local](#), lo que puede acelerar tu proceso de desarrollo, ya que te permite usar Zonas Locales o registrar tu propio metal como parte de tu flota.

Pasos para la implementación

- Utiliza infrastructure-as-code herramientas como AWS CloudFormation Terraform para realizar despliegues repetibles, lo que te permitirá personalizar y escalar rápidamente las ubicaciones de alojamiento de juegos en función de las necesidades de los jugadores.
- Evalúa las mejoras en la experiencia de los jugadores, las prioridades de la población de jugadores y los impactos posteriores, como los tiempos de emparejamiento, al añadir o eliminar ubicaciones de alojamiento de juegos.
- Utilice AWS Local Zones, Outposts u opciones híbridas como ECS Anywhere, EKS Anywhere o GameLift Anywhere para optimizar la infraestructura sensible a la latencia y satisfacer diversas necesidades de implementación.

Desarrollo iterativo

GAMEPERF03: ¿Cómo puedes usar Amazon GameLift para mejorar el rendimiento del desarrollo iterativo?

Amazon GameLift proporciona un end-to-end flujo de trabajo para el desarrollo y las pruebas de rendimiento de tu juego en un entorno de pruebas local.

Prácticas recomendadas

- [GAMEPERF03-BP01 Usa GameLift Amazon Anywhere y un kit de herramientas de prueba GameLift](#)
- [GAMEPERF03-BP02 Pruebe el rendimiento y la escalabilidad de los servidores de juegos](#)
- [GAMEPERF03-BP03 Optimiza la utilización de los recursos de los contenedores GameLift](#)

GAMEPERF03-BP01 Usa GameLift Amazon Anywhere y un kit de herramientas de prueba GameLift

Para mejorar la eficiencia del rendimiento mediante un proceso de desarrollo iterativo, utilice Amazon GameLift Anywhere junto con el Amazon GameLift Testing Toolkit para establecer un entorno de pruebas integral.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Este enfoque permite una iteración rápida, una recopilación de datos eficiente y un análisis detallado del rendimiento. Los pasos clave incluyen:

Cree un entorno de pruebas

Utilice Amazon GameLift Anywhere para configurar un entorno de pruebas local o basado en la nube. Esta configuración elimina la necesidad de cargar cada iteración de compilación del servidor de juegos a una flota gestionada, lo que reduce el tiempo de activación.

Integrar Amazon GameLift Testing Toolkit

Incorpore el Amazon GameLift Testing Toolkit en su flujo de trabajo de desarrollo. El kit de herramientas proporciona scripts, herramientas y bibliotecas para visualizar la GameLift infraestructura de Amazon, lanzar reproductores virtuales e iterar conjuntos de FlexMatch reglas con el FlexMatch simulador. Simplifica la integración y la administración de GameLift los recursos de Amazon, lo que le permite automatizar las tareas comunes y recopilar los datos necesarios para el análisis del rendimiento.

Ciclos rápidos de construcción y prueba

Actualice rápidamente la flota de pruebas con nuevas versiones, iníciela y comience las pruebas. Esto facilita un build-test-repeat ciclo rápido, lo que permite a los desarrolladores validar varios aspectos de la experiencia de los jugadores del juego, incluidas las interacciones multijugador.

Pruebas exhaustivas

Pon a prueba la integración de tu servidor de juegos con el SDK GameLift del servidor de Amazon, las interacciones de los servicios de backend, las configuraciones de emparejamiento y otras funciones de GameLift alojamiento. Utiliza el kit de herramientas de GameLift pruebas para automatizar las pruebas y recopilar métricas de rendimiento detalladas, asegurándote de que los componentes del juego funcionen juntos a la perfección.

Analiza los datos de rendimiento

Usa los datos recopilados por el kit de herramientas de GameLift pruebas para analizar los cuellos de botella en el rendimiento y optimizar tu servidor de juegos. El kit de herramientas ayuda a realizar un seguimiento de las métricas clave, identificar problemas y tomar decisiones basadas en datos para mejorar la eficiencia del rendimiento.

Al incorporar Amazon GameLift Anywhere y el kit de herramientas de GameLift pruebas en su proceso de desarrollo iterativo, puede mejorar considerablemente la eficiencia del rendimiento mediante pruebas rápidas, comprobaciones de integración exhaustivas y análisis detallados del rendimiento.

Pasos para la implementación

- Utilice Amazon GameLift Anywhere para crear un entorno de pruebas, reducir el tiempo de activación de las compilaciones de servidores de juegos y permitir una iteración rápida.
- Integre el Amazon GameLift Testing Toolkit para automatizar las tareas de prueba, simular jugadores y validar FlexMatch las configuraciones durante el desarrollo.

- Recopile y analice los datos de rendimiento con el kit de herramientas de GameLift pruebas para identificar los cuellos de botella, optimizar los servidores de juegos y mejorar la eficiencia del rendimiento.

GAMEPERF03-BP02 Pruebe el rendimiento y la escalabilidad de los servidores de juegos

Para probar el rendimiento y la escalabilidad de tus servidores de juegos, implementa un marco de pruebas sólido con las GameLift funciones de Amazon y el kit de herramientas de GameLift pruebas.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Las prácticas clave incluyen:

Pruebas iterativas

Utilice una flota de Amazon GameLift Anywhere para crear un entorno alojado en la nube en el que pueda crear y probar componentes de juegos de forma iterativa. Este entorno debe reflejar las condiciones de alojamiento del mundo real, lo que permite realizar pruebas realistas de rendimiento y escalabilidad.

Pruebas de integración de servidores de juegos

Pruebe la integración de su servidor de juegos con el SDK del servidor de Amazon GameLift, lo que incluye iniciar nuevas sesiones de juego y realizar un seguimiento de los eventos de las sesiones de juego mediante nuestro AWS CLI kit de herramientas de GameLift prueba. Esto verifica que el servidor del juego funcione correctamente en el GameLift entorno.

Usa el kit GameLift de herramientas de pruebas para automatizar las pruebas y recopilar métricas de rendimiento detalladas. El kit de herramientas te permite visualizar la GameLift infraestructura, lanzar reproductores virtuales para realizar pruebas de carga e iterar conjuntos de FlexMatch reglas con el simulador. FlexMatch Es particularmente útil para escalar las tareas de ECS Fargate, que simulan las sesiones de los jugadores mediante la creación de numerosas sesiones de juego simultáneas para poner a prueba la infraestructura del servidor.

Pruebas de escalabilidad

Experimente con diseños de colas de sesiones de juego, flotas con varias ubicaciones, flotas puntuales y bajo demanda y varios tipos de instancias. Pruebe las opciones de ubicación de las sesiones de juego, las políticas de latencia y la configuración de priorización de la flota. Configura el escalado de capacidad para satisfacer la demanda de los jugadores y comprueba que el sistema puede gestionar la carga prevista en diferentes condiciones.

Pasos para la implementación

- Utilice Amazon GameLift Anywhere para configurar un entorno de pruebas realista para realizar pruebas iterativas de rendimiento y escalabilidad.
- Pruebe la integración del servidor de juegos con el SDK GameLift del servidor, lo que facilitará la correcta administración de las sesiones y el seguimiento de los eventos dentro del GameLift entorno.
- Realiza pruebas de escalabilidad con el kit de herramientas de GameLift pruebas, simulando la carga de jugadores, probando las colas de sesiones y validando la escalabilidad de la flota, las políticas de latencia y los ajustes de priorización.

GAMEPERF03-BP03 Optimiza la utilización de los recursos de los contenedores GameLift

Para optimizar la utilización de los recursos de GameLift los contenedores, diseñe su flota de contenedores de forma eficaz y establezca límites de recursos precisos.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Las pautas clave incluyen:

- Diseño de grupos de contenedores: organice su software en grupos de contenedores. El contenedor principal debe incluir la aplicación del servidor de juegos y el Amazon GameLift Agent. Utiliza contenedores sidecar como software adicional para gestionar las dependencias y establecer límites específicos de los contenedores para el uso de memoria y CPU.
- Establezca límites de recursos: para cada grupo de contenedores, determine los recursos de memoria y CPU necesarios. Establezca límites opcionales para los contenedores individuales a fin de verificar que tengan recursos reservados, pero que también puedan superarlos si hay recursos adicionales disponibles. Esto ayuda a evitar la contención de recursos y las posibles fallas en los contenedores.

- Grupo de contenedores daemon: considere la posibilidad de utilizar un grupo de contenedores daemon para procesos en segundo plano o de supervisión que no necesiten escalarse con el grupo de contenedores principal. Esto verifica que las tareas esenciales en segundo plano se gestionen de forma eficiente sin que ello afecte a los procesos principales del servidor del juego.

Pasos para la implementación

- Diseña grupos de contenedores con un contenedor principal para el servidor del juego y el GameLift agente, y sidecars para gestionar las dependencias, con límites específicos de memoria y CPU.
- Establece límites de recursos para cada grupo de contenedores a fin de reservar los recursos necesarios y, al mismo tiempo, permitir un uso controlado de los recursos para evitar la contención.
- Usa un grupo de contenedores daemon para tareas en segundo plano o de monitoreo, asegurándote de que funcionan de manera eficiente sin afectar a los procesos del servidor principal del juego.

Computación y hardware

GAMEPERF04: ¿Cómo se evita que las sesiones de juego afecten a los jugadores que utilizan la misma instancia de servidor de juegos?

Una vez que el servidor de juegos esté funcionando AWS, tendrás que supervisar su rendimiento para ofrecer una experiencia de calidad a los jugadores, independientemente del uso de los recursos, la computación subyacente o la saturación.

Prácticas recomendadas

- [GAMEPERF04-BP01 Supervisa los procesos del servidor de juegos para detectar problemas](#)
- [GAMEPERF04-BP02 Pon a prueba el rendimiento de tu servidor de juegos con escenarios de juego simulados y reales](#)

GAMEPERF04-BP01 Supervisa los procesos del servidor de juegos para detectar problemas

Puede ejecutar varios procesos de servidor de juegos por instancia para utilizar de manera eficiente los recursos de sus instancias de servidor de juegos. Si es así, diseña tu arquitectura de forma que un proceso de servidor de juegos individual que aloje una sesión de juego no pueda afectar negativamente a otras sesiones de juego alojadas en la misma instancia. Usa las métricas para entender cómo la ubicación del juego y el tipo de modo de juego pueden afectar al rendimiento de las instancias del servidor de juegos. Incorpora una combinación de procesos de baja carga (tutorial en el lobby, tienda o para un jugador) y de alta carga (jugabilidad clasificatoria, multijugador o de alta habilidad) para evitar que la instancia del servidor del juego se vea afectada.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Supervisa la experiencia del jugador mediante métricas del lado del cliente y del lado del servidor mediante la recopilación de datos telemétricos para medir el tiempo de ping y la fluctuación, las caídas de fotogramas, el tiempo de respuesta de la API, los errores y la finalización correcta del ciclo de juego. Correlaciona las fechas de estos eventos con los problemas de asistencia a los jugadores y los registros del servidor para identificar los cuellos de botella en el rendimiento. Se pueden utilizar herramientas como [Dtrace](#), [ftrace](#), [uperf](#) y eBPF para investigar y analizar en profundidad el rendimiento [del](#) sistema.

Implemente la supervisión de los recursos limitados disponibles en las instancias de sus servidores de juegos para poder generar alertas cuando los procesos individuales del servidor de juegos superen los límites presupuestarios de recursos predeterminados. Cuando se superen los umbrales, puede que desees configurar el software del servidor de juegos para que transfiera las sesiones pertinentes del sistema y del servidor de juegos a un almacenamiento duradero, como una solución de registro central, para que los ingenieros del servidor de juegos puedan investigar este comportamiento. Además, la instancia del servidor de juegos debe estar configurada para informar de las métricas de cada uno de los procesos del servidor de juegos que se ejecutan en la instancia, de modo que pueda supervisar estos procesos individuales del servidor de juegos, además de las métricas generales de la instancia del servidor de juegos.

Por ejemplo, GameLift proporciona métricas para [monitorear las sesiones de juego](#), que se pueden complementar con métricas personalizadas específicas del juego y registros recopilados mediante el [Amazon CloudWatch Agent](#) que puedes configurar en tu instancia de servidor de juegos. Sus

métricas se pueden ver CloudWatch o exportar a otras herramientas, como [Amazon Managed Grafana](#), que está integrada con Single Sign-On para facilitar el acceso a las métricas a los usuarios que quizás no tengan acceso a la Consola de administración. Consulta las siguientes prácticas recomendadas para [gestionar los registros y las métricas con Amazon GameLift](#), que también ofrece soporte para ver los [registros de las sesiones de juego](#) individuales.

Pasos para la implementación

- Ejecuta varios procesos de servidor de juegos por instancia y mezcla modos de juego con carga alta y baja para evitar problemas y comprobar un uso equilibrado de los recursos.
- Supervisa las métricas del lado del cliente y del servidor, como el ping, la fluctuación, las caídas de fotogramas y los tiempos de respuesta de las API, y correlaciónalas con los registros del servidor y los problemas reportados por los jugadores para identificar los cuellos de botella.
- Configura la supervisión de recursos para cada proceso del servidor de juegos, genera alertas en caso de que se superen los umbrales y guarda los registros en un almacenamiento duradero para analizarlos con herramientas como CloudWatch Grafana gestionada por Amazon.

GAMEPERF04-BP02 Pon a prueba el rendimiento de tu servidor de juegos con escenarios de juego simulados y reales

Realiza pruebas de rendimiento y evalúa varios escenarios de juego para determinar si el proceso del servidor del juego gestiona adecuadamente la utilización de los recursos fijos, como la memoria de la EC2 instancia, la CPU y el ancho de banda de la red.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Al crear pruebas de juego simuladas con bots que puedan reflejar las rutas y comportamientos de juego habituales de tus jugadores, podrás determinar la forma en que tu servidor de juegos gestiona esta situación en distintos escenarios de uso. Por ejemplo, puedes implementar una solución, como las [pruebas de carga distribuidas](#), AWS que puedas personalizar para ejecutar simulaciones de clientes de juegos o compilaciones de clientes de juegos para generar escenarios de juego. Realiza pruebas de juego internas y utiliza equipos de control de calidad para poner a prueba las distintas características de tu juego, de modo que puedas tener la confianza de que tu juego está diseñado para funcionar de forma óptima. [AWS Device Farm](#) se puede usar para realizar pruebas móviles y web para tus juegos de iOS, Android y navegador en varios tipos de dispositivos.

Pasos para la implementación

- Realiza pruebas de rendimiento con bots que simulen los comportamientos habituales de los jugadores para evaluar la utilización de los recursos del servidor de juegos en diferentes escenarios.
- Usa soluciones como Distributed Load Testing AWS para personalizar y simular escenarios de juego para las pruebas de stress.
- Realiza pruebas de juego internas y utiliza herramientas como las pruebas de juegos AWS Device Farm para dispositivos móviles y de navegador en varios dispositivos.

Selección de cómputo

GAMEPERF05: ¿Cómo seleccionas la solución informática adecuada para tu juego?

El rendimiento informático varía según el tamaño y la familia de las instancias. Resulta beneficioso utilizar varias opciones de procesamiento que provengan de grupos de capacidad independientes. Desarrolle una estrategia de composición de la flota que dé preferencia al rendimiento, pero que incluya la suficiente diversidad para evitar errores de capacidad insuficiente.

Prácticas recomendadas

- [GAMEPERF05-BP01 Compara el rendimiento de tus juegos en varios tipos de cómputo](#)
- [GAMEPERF05-BP02 non-latency-sensitive Transfiere las tareas informáticas a flujos de trabajo asíncronos](#)

GAMEPERF05-BP01 Compara el rendimiento de tus juegos en varios tipos de cómputo

En cuanto a las cargas de trabajo de los servidores de juegos, no existe un enfoque único para identificar la solución informática óptima para alojar el servidor de juegos. Una estrategia habitual para realizar una evaluación comparativa de los servidores de juegos consiste en empezar con instancias EC2 «c» optimizadas para la informática, ya que esta familia de instancias proporciona un alto rendimiento para cargas de trabajo que requieren un uso intensivo de recursos computacionales.

Como alternativa, si el juego requiere una cantidad importante de memoria para implementar funciones específicas, las instancias con memoria optimizada pueden ser las más adecuadas.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Si tu carga de trabajo utiliza muchos recursos de red, considera la posibilidad de implementar instancias optimizadas para la red, lo que normalmente se indica con una «n» en el nombre de la instancia, y evita los tipos de instancia fragmentables («t»), ya que una vez agotados los créditos, el rendimiento disminuirá. Los juegos son sensibles a la latencia y a la pérdida de paquetes, por lo que se recomienda utilizar EC2 una red mejorada para mejorar el rendimiento de la red de los servidores de juegos. [Las redes mejoradas utilizan la I/O virtualización de raíz única \(SR-IOV\) para proporcionar capacidades de red de alto rendimiento en los tipos de instancias compatibles.](#) El SR-IOV es un método de virtualización de dispositivos que proporciona un mayor I/O rendimiento y un menor uso de la CPU en comparación con las interfaces de red virtualizadas tradicionales. Las redes mejoradas proporcionan un mayor ancho de banda, un rendimiento superior de paquetes por segundo (PPS) y menores latencias entre instancias de manera constante. [Las redes mejoradas con Elastic Network Adapter están disponibles para los tipos de EC2 instancias más recientes y es importante actualizarlas periódicamente para aprovechar las mejoras de rendimiento de las instancias más nuevas y las mejoras del hipervisor Nitro.AWS](#)

Si tu juego funciona de forma similar en varios tipos de EC2 instancias, deberías considerar usar varios tipos de instancias para alojar tus servidores de juegos. Supervisa el rendimiento a lo largo del tiempo y optimiza aún más una vez que hayas organizado suficientes sesiones de juegos de producción para poder identificar las tendencias de rendimiento. Recuerda que tus requisitos informáticos pueden cambiar a medida que añadas nuevas funciones al juego que requieran una asignación de recursos diferente. Puede [configurar los grupos de EC2 Auto Scaling](#) para que usen varios tipos de instancias, o puede usar grupos de Auto Scaling independientes para alojar instancias de servidores de juegos que ejecuten tipos de instancias independientes, lo que puede facilitar la administración de la correlación y la agregación de métricas.

Evalúa el rendimiento de tu juego en distintos tipos de procesadores, como las instancias basadas en Intel, las instancias basadas en AMD y las instancias Graviton basadas en ARM. Unreal Engine 5.1.1 o una [versión posterior puede compilar servidores de juegos para Graviton y mejorar la relación precio-rendimiento de tu juego.](#) Realiza pruebas de barrido y saturación en distintos tamaños dentro de cada familia para determinar el punto óptimo en el que la utilización y el rendimiento son consistentes.

También deberías comparar cómo se ve afectado el rendimiento de tu juego cuando se aloja mediante contenedores y funciones Lambda. Para los casos de uso en los que no se requieran procesos de servidor de juegos de larga duración, como los juegos asíncronos y los servicios de backend de juegos, considere la posibilidad de utilizar una arquitectura sin servidor con Lambda que pueda simplificar la administración y las operaciones de los equipos de operaciones de juego, además de permitirle implementar su juego de manera más rápida a nivel mundial para muchos. Regiones de AWS Para conocer las mejores prácticas sin servidor, consulte [Serverless Applications Lens: Well-Architected Framework](#).

Pasos para la implementación

- Compare los servidores de juegos con instancias «c» optimizadas para la computación para cargas de trabajo con un uso intensivo de la CPU, instancias optimizadas para la memoria para tareas con uso intensivo de memoria e instancias «n» optimizadas para la red para lograr un alto rendimiento de la red.
- Utilice redes mejoradas con Elastic Network Adapter (ENA) en las instancias compatibles para mejorar el rendimiento de la red, reducir la latencia y aumentar las velocidades de procesamiento de paquetes.
- Evalúe y pruebe varios tipos de instancias, procesadores (Intel, AMD, Graviton) y opciones de alojamiento de contenedores o Lambda, ajustando las soluciones informáticas a medida que evolucionan las funciones de los juegos.

Para obtener más información, consulte [Elija la estrategia de procesamiento adecuada para sus servidores de juegos globales](#).

GAMEPERF05-BP02 non-latency-sensitive Transfiere las tareas informáticas a flujos de trabajo asíncronos

A la hora de optimizar el rendimiento del juego, es importante tener en cuenta que solo algunas de las interacciones entre el cliente y el backend del juego deben realizarse de forma sincrónica. Deberías considerar cada función desde el punto de vista de la experiencia del jugador y determinar si determinadas interacciones requieren comunicaciones sincrónicas, que generan bloqueos y consumen muchos recursos, o si esas funciones se pueden implementar de forma asíncrona. Cuando implementes llamadas de red, usa un enfoque asíncrono y sin bloqueo. Además, el backend del juego también debe estar configurado para funcionar de manera eficiente, descargando las tareas en colas y priorizando las respuestas rápidas a los clientes siempre que sea posible.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Por ejemplo, la actualización de una tabla de clasificación al final de la sesión de un jugador se puede implementar de forma asíncrona para que el cliente no tenga que esperar a que se complete la actualización de la tabla de clasificación. En su lugar, impleméntelo de forma asíncrona en el cliente del juego y considere la posibilidad de diseñar su servicio de backend para incluir este tipo de operaciones en colas, como Amazon SQS. Con esta arquitectura, configura tu backend para que acepte la solicitud, ponla en cola en SQS, lo que ayuda a almacenar los mensajes de forma duradera para su procesamiento asíncrono, y responde rápidamente al cliente. Cuando se complete la actualización de la tabla de clasificación, el backend puede enviar una actualización al cliente del juego para actualizar la visión del jugador sobre la tabla de clasificación.

Como alternativa, el jugador puede simplemente visitar la pantalla de clasificación de tu juego para recuperar los datos más recientes, lo que puede enviar una solicitud web a tu backend para recuperar los datos más recientes de la memoria caché.

Pasos para la implementación

- Determina si las interacciones entre el cliente y el backend requieren una comunicación sincrónica; implementa enfoques asíncronos y sin bloqueos siempre que sea posible para optimizar el uso de los recursos.
- Utilice Amazon SQS para deshacerse de las tareas no críticas, como las actualizaciones de las tablas de clasificación.
- Permita que el cliente obtenga datos actualizados de forma asíncrona, por ejemplo, recuperando los datos más recientes de la tabla de clasificación a pedido o mediante actualizaciones en segundo plano.

Recursos

- [Descripción de la mensajería asíncrona para microservicios](#)
- [Lambda: uso de integraciones de servicios y procesamiento asíncrono](#)

Administración de datos

GAMEPERF06: ¿Cómo se gestionan y analizan de forma eficiente los registros de los servidores de juegos y se almacenan diferentes tipos de datos de juego para lograr un rendimiento óptimo?

Los juegos pueden contener datos de los jugadores, registros de juegos y registros del servidor, que deben estar separados unos de otros siempre que sea posible. Centralizar la ingesta de registros y la gestión del ciclo de vida puede beneficiar a tus equipos de juego, ya que proporciona información sobre lo que ocurre en el juego y en el servidor.

Prácticas recomendadas

- [GAMEPERF06-BP01 Centraliza la recopilación y el almacenamiento de registros](#)
- [GAMEPERF06-BP02 Clasifica y almacena los datos del juego en función de los patrones de acceso](#)
- [GAMEPERF06-BP03 Habilita el formateo de registros y el procesamiento por lotes eficientes](#)
- [GAMEPERF06-BP04 Implemente políticas de rotación y retención de registros](#)
- [GAMEPERF06-BP05 Usa herramientas de monitorización y visualización](#)

GAMEPERF06-BP01 Centraliza la recopilación y el almacenamiento de registros

Implemente una solución centralizada de recopilación y almacenamiento de registros para recopilar registros de las instancias del servidor de juegos y GameLift.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Usa servicios como Amazon CloudWatch Logs para recopilar, monitorear y almacenar datos de registro de tus servidores e GameLift instancias de juegos. CloudWatch Logs proporciona una solución escalable y totalmente gestionada para la gestión de registros, que facilita el almacenamiento y la recuperación eficientes de los datos de registro sin afectar al rendimiento del servidor de juegos. Si utilizas el [agente CloudWatch Logs](#), ten en cuenta los distintos tipos de instalación y opciones de configuración, como el tamaño del lote y la duración del búfer, para

minimizar el impacto en el servidor del juego. Considera que las instancias del servidor de juegos son efímeras y reduce la dependencia del registro localizado siempre que sea posible. Establezca una política centralizada para la implementación de las [mejores prácticas de registro](#).

Pasos para la implementación

- Utilice Amazon CloudWatch Logs para recopilar, supervisar y almacenar datos de registro de instancias de servidores de juegos y GameLift, así, facilitar la administración de registros centralizada y escalable.

GAMEPERF06-BP02 Clasifica y almacena los datos del juego en función de los patrones de acceso

Clasifica los datos del juego en diferentes tipos según sus patrones de acceso y requisitos de almacenamiento.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Entre las categorías más comunes se incluyen los datos de los jugadores, las partidas guardadas, el almacenamiento persistente del mundo y los datos analíticos.

Pasos para la implementación

Utilice las soluciones de almacenamiento adecuadas para cada tipo de datos a fin de optimizar el rendimiento y la rentabilidad:

- Datos de los jugadores: utilice Amazon DynamoDB, una base de datos NoSQL rápida y escalable, para almacenar los perfiles, las preferencias y los datos de progresión de los jugadores. El acceso de baja latencia y las capacidades de escalado automático de DynamoDB permiten recuperar y actualizar los datos de los jugadores de forma eficiente.
- Partidas guardadas: utilice Amazon S3 para almacenar partidas guardadas y puntos de control. S3 ofrece una gran durabilidad y escalabilidad para almacenar grandes cantidades de datos guardados en partidas. Considera la posibilidad de utilizar S3 Transfer Acceleration o Amazon CloudFront para subir y descargar partidas guardadas con mayor rapidez.
- Almacenamiento mundial persistente: para juegos con estados mundiales persistentes o datos de juegos compartidos, considere la posibilidad de utilizar Amazon DynamoDB, Amazon o

ElastiCache Amazon MemoryDB. ElastiCache y MemoryDB proporcionan un almacén de valores clave en memoria, mientras que DynamoDB es una base de datos NoSQL respaldada por SSD. Estos servicios proporcionan un acceso rápido a los datos almacenados, lo que reduce el tiempo que tarda el servidor del juego en guardar el estado del juego, lo que mejora el rendimiento general del proceso.

- Datos de análisis: utilice Amazon Managed Streaming for Apache Kafka o Kinesis Data Streams para incorporar flujos de datos de sus productores de datos de juegos. Amazon Managed Service for Apache Flink se puede utilizar para la transformación y el análisis en tiempo real y se puede enviar a Amazon Data Firehose para su procesamiento y entrega a lagos de datos, almacenes y servicios de análisis de back-end. La [guía para Game Analytics Pipeline AWS](#) ilustra cómo funcionan juntos los servicios para proporcionar análisis por lotes y casi en tiempo real.

GAMEPERF06-BP03 Habilita el formateo de registros y el procesamiento por lotes eficientes

Configura los procesos de tu servidor de juegos para que generen registros estructurados y en un formato que se pueda analizar, como JSON.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Implemente técnicas de procesamiento por lotes de registros para minimizar la frecuencia de las transferencias de datos de registro desde sus servidores de juegos al almacenamiento de registros centralizado. La agrupación de registros por lotes reduce la sobrecarga de la red y mejora el rendimiento del servidor de juegos. Utilice los registros detallados o de nivel de depuración como excepción y no como valores predeterminados, ya que pueden suponer una penalización en el rendimiento y los costes que debe evitarse siempre que sea posible.

GAMEPERF06-BP04 Implemente políticas de rotación y retención de registros

Establezca políticas de rotación y retención de registros para gestionar el crecimiento de los datos de registro y optimizar la utilización del almacenamiento.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: bajo

Guía para la implementación

Configura tus servidores de juegos para que roten automáticamente los registros según el tamaño o los intervalos de tiempo. Defina políticas de retención de CloudWatch registros en Amazon Logs para archivar o eliminar automáticamente los datos de registro antiguos que ya no se necesitan para el análisis activo o la solución de problemas.

GAMEPERF06-BP05 Usa herramientas de monitorización y visualización

Utilice las herramientas de supervisión y visualización para obtener información sobre el rendimiento de su servidor de juegos e identificar las oportunidades de optimización.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Usa Amazon CloudWatch para monitorear las métricas clave y configurar alarmas para recibir notificaciones proactivas. Utiliza herramientas como Amazon Managed Service for Prometheus y Amazon Managed Grafana para recopilar, consultar y visualizar las métricas de los servidores y la infraestructura de tus juegos. Cree paneles informativos para realizar un seguimiento del rendimiento, identificar los cuellos de botella y realizar optimizaciones basadas en los datos.

Redes y entrega de contenido

GAMEPERF07: ¿Cómo diseñas tu servicio de matchmaking para optimizar el rendimiento?

Es importante tener en cuenta la habilidad de los jugadores, la calidad del proveedor de servicios de Internet (ISP) y la distribución de la población de jugadores como dimensiones del ajuste del rendimiento. Las sesiones de juego se pueden colocar en servidores ubicados estratégicamente para nivelar las condiciones de juego y organizar un juego limpio.

Prácticas recomendadas

- [GAMEPERF07-BP01 Define los umbrales de latencia de red para tu juego](#)
- [GAMEPERF07-BP02 Ejecuta un servicio de emparejamiento independiente para cada modo de juego y región de alojamiento del juego](#)

- [GAMEPERF07-BP03 Supervisa regularmente el rendimiento del matchmaking](#)
- [GAMEPERF07-BP04 Supervisa regularmente el rendimiento de la red](#)
- [GAMEPERF07-BP05 Utilice la tecnología de aceleración de red para mejorar el rendimiento en Internet](#)

GAMEPERF07-BP01 Define los umbrales de latencia de red para tu juego

Al desarrollar un juego multijugador, comprueba que la infraestructura de juego no provoque una latencia innecesaria para los jugadores. Si tu juego es sensible a la latencia de la red, debes establecer umbrales de latencia en tu lógica de emparejamiento para priorizar la ubicación de los jugadores en las sesiones del servidor del juego que estén alojadas en ubicaciones cercanas del servidor de juegos o Regiones de AWS que cumplan tu objetivo de ofrecer una experiencia de juego ideal.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

En muchos juegos sensibles a la latencia, es común instruir a los clientes del juego para que hagan ping a cada una de las ubicaciones de la infraestructura del juego para recopilar datos de rendimiento, como la latencia de la red, la fluctuación y la pérdida de paquetes, y reportar estos datos al backend de recopilación de métricas para que puedan analizarse. Al unir jugadores en las sesiones de juego, puedes configurar tu juego para que incorpore la latencia de red percibida por el cliente del juego a la infraestructura del servidor de juego como una de las entradas utilizadas en la lógica del servicio de emparejamiento.

GAMEPERF07-BP02 Ejecuta un servicio de emparejamiento independiente para cada modo de juego y región de alojamiento del juego

Si tu juego ofrece varios modos de juego entre los que los jugadores pueden elegir, debes separar los sistemas de emparejamiento para cada uno de ellos para poder ajustar de forma independiente el rendimiento de cada modo de juego en función de sus requisitos únicos y reducir la escasez de recursos. Es probable que cada modo de juego tenga requisitos únicos en cuanto a la latencia aceptable, el tamaño de las partidas y otras lógicas de emparejamiento personalizadas y específicas del juego. También es probable que atraigan a diferentes tipos de jugadores. Ejecuta el servicio de emparejamiento de cada modo de juego como una implementación de software independiente para que puedas probar el rendimiento y operar los modos de juego de forma independiente.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Por ejemplo, puede ejecutarlas como funciones Lambda independientes para cada modo de juego o puede utilizarlas como despliegues de servicios independientes basados en contenedores.

Implementa tus servicios de búsqueda de partidas en varias regiones cercanas a las ubicaciones de tus servidores de juegos. El tráfico de jugadores tomará muchas rutas, por lo que es importante que el servicio de emparejamiento mantenga un perfil de up-to-date latencia en varias ISPs para mejorar la eficiencia de las sesiones de juego con baja latencia. GameLift FlexMatch proporciona una guía adicional para seleccionar regiones para los emparejadores e incluye la posibilidad de integrar tus emparejadores con colas de sesiones de juego de [varias regiones](#).

GAMEPERF07-BP03 Supervisa regularmente el rendimiento del matchmaking

Una de las formas más notables de optimizar el rendimiento de un juego para los jugadores es reducir el tiempo que deben esperar antes de poder entrar en una sesión de juego. Los tiempos de espera prolongados pueden hacer que los jugadores pierdan interés y provoquen deserción, por lo que es importante tener esto en cuenta a la hora de diseñar tu solución de matchmaking.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Cuando diseñes la configuración de emparejamiento para tu juego, crea reglas que determinen las condiciones que se deben aplicar para formar una partida. Deberías tener en cuenta el impacto que estas reglas tendrán en el rendimiento del sistema, especialmente en los tiempos de espera de los jugadores. Antes de introducir cambios en la implementación del sistema de emparejamiento, como añadir nuevas condiciones o filtros de emparejamiento, pruébalo con antelación o plantéate la posibilidad de aplicar este cambio de forma gradual a una pequeña muestra de jugadores a modo de canalización, o bien haz una A/B prueba para recopilar estadísticas de rendimiento antes de introducir este cambio en toda la población de jugadores.

Configura tu servicio de emparejamiento para generar registros detallados que te permitan entender las condiciones o reglas que se aplicaron a cada solicitud de emparejamiento. Esto ayuda a revisar y ajustar la implementación del emparejamiento según sea necesario.

Por ejemplo, [Amazon GameLift FlexMatch](#) proporciona un servicio de búsqueda de pareja totalmente gestionado que se puede utilizar como un servicio independiente con el alojamiento de su propio servidor de juegos o con servidores de juegos alojados en Amazon. GameLift FlexMatch puede generar notificaciones de eventos para Amazon EventBridge, consulta [Configurar notificaciones de FlexMatch eventos](#). Utilice Amazon Simple Notification Service (Amazon SNS) para recibir datos de emparejamiento en formato JSON, lo que le permitirá procesar y almacenar automáticamente esta información para analizarla y mejorar el rendimiento del emparejamiento.

Configura métricas para hacer un seguimiento del tiempo que tarda tu servicio de búsqueda de parejas en encontrar una sesión de juego adecuada para los jugadores. Revisa las métricas de duración del emparejamiento con regularidad y correlaciona estos tiempos con el comportamiento de los jugadores y la opinión de la comunidad. Usa estos datos para desarrollar umbrales adecuados para los tiempos de espera de los emparejamientos que puedas incluir en la configuración de tus reglas de emparejamiento.

Por ejemplo, Amazon GameLift FlexMatch brinda soporte para definir los tiempos de espera de las solicitudes de emparejamiento, así como para crear reglas de emparejamiento que pueden [permitir que los requisitos se relajen con](#) el tiempo. Esta función te permite crear emparejamientos adaptables para facilitar la creación de partidas y colocar a los jugadores en las sesiones de juego cuando sea difícil encontrarlas.

GAMEPERF07-BP04 Supervisa regularmente el rendimiento de la red

Para los juegos competitivos, es importante tener una experiencia de jugador coherente.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Una partida fiable de 50 ms para una base de jugadores más grande es más justa y divertida que una partida en la que un jugador tiene un ping de 10 ms y otro que tiene un ping de 70 ms. Los cambios en las rutas de los proveedores de servicios de Internet pueden afectar a una parte de la población de jugadores, por lo que tu sistema de búsqueda de partidas tendrá que adaptarse.

[Amazon CloudWatch Network Monitoring](#) ayuda a determinar si el problema está relacionado con tu juego o con el proveedor de Internet del jugador.

Pasos para la implementación

- Utilice Amazon Cloudwatch Network Monitoring para realizar un seguimiento del rendimiento de la red e identificar los problemas de enrutamiento.

- Usa los registros de flujo de la VPC para identificar patrones de tráfico anormales o paquetes descartados, lo que puede indicar congestión de la red, problemas con el ISP o errores de configuración que afectan a la latencia del reproductor.

GAMEPERF07-BP05 Utilice la tecnología de aceleración de red para mejorar el rendimiento en Internet

Además de acercarse físicamente a los jugadores la infraestructura de juego sensible a la latencia, también puedes mejorar la experiencia de los jugadores optimizando el rendimiento de la red para tu juego. AWS utiliza el protocolo BGP para influir en el [enrutamiento de Internet](#) y utilizar la ruta más rápida hacia nuestra red fronteriza desde los proveedores de servicios de Internet. Si utilizas tu propia red y necesitas más control y visibilidad sobre el comportamiento de enrutamiento y la publicidad de BGP, puedes usar el [peering privado o redirigir](#) el tráfico de Internet Direct Connect a tu juego en el que se esté ejecutando. AWS

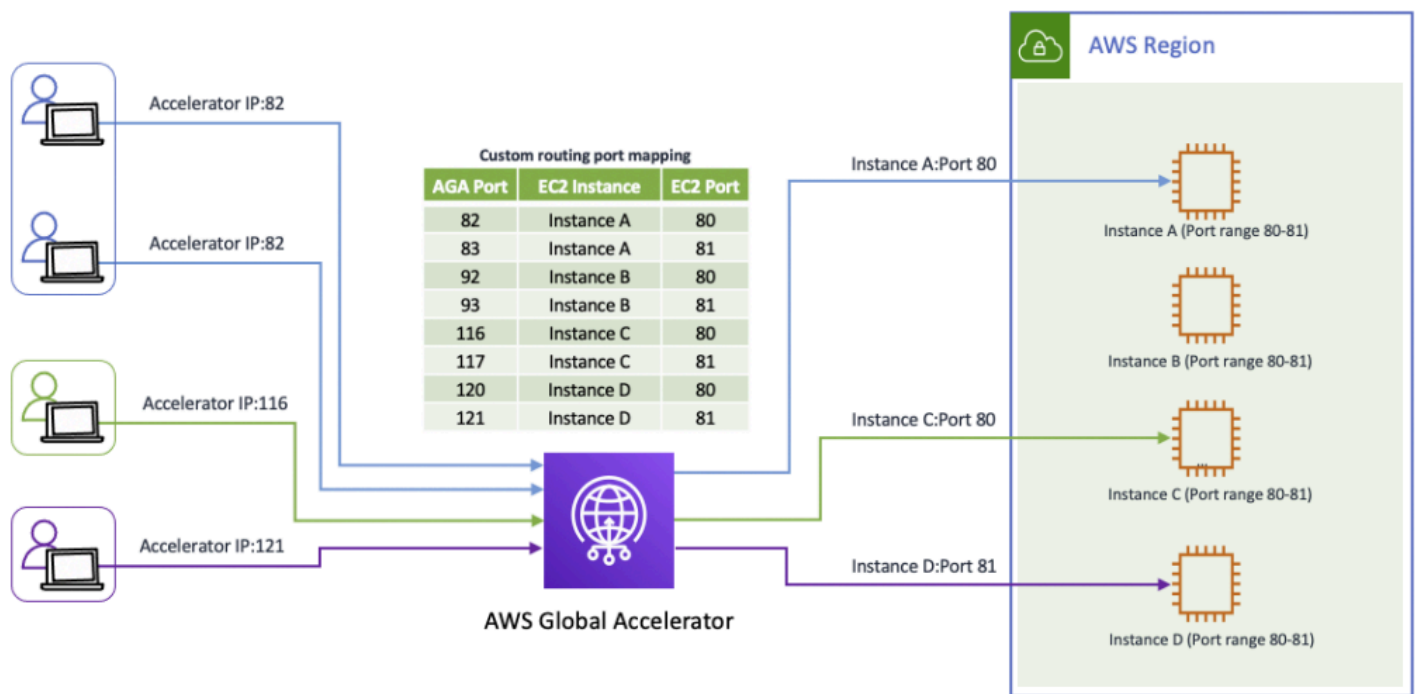
Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Tenga en cuenta la siguiente arquitectura de referencia para mejorar el rendimiento y la capacidad de respuesta de Internet.

Rendimiento de red mejorado para juegos con Global Accelerator

Si busca una solución de enrutamiento de red totalmente gestionada, [AWS Global Accelerator](#) mejora el rendimiento de la red de su aplicación mediante la red AWS global, que se puede utilizar para acelerar el tráfico de juegos, el chat de voz y el tráfico de mensajería en tiempo real, así como otras aplicaciones sensibles a la latencia, al tiempo que proporciona una rápida conmutación por error a los servidores de juegos. Los [aceleradores de enrutamiento personalizados de Global Accelerator](#) se pueden integrar con su servicio de emparejamiento para proporcionar un enrutamiento determinista de varios jugadores a la misma sesión de juego mediante direcciones IP y puertos estáticos de transmisión automática.



Es posible que tus equipos de desarrollo de juegos estén distribuidos por todo el mundo y necesiten un acceso eficiente al contenido o los activos compartidos. Para mejorar el rendimiento del contenido compartido almacenado en los depósitos de Amazon S3, puede configurar la replicación bidireccional de los datos en todas las regiones mediante la replicación entre regiones de [S3](#) para que los usuarios puedan acceder a los datos de los depósitos más cercanos a ellos. Para simplificar este patrón de acceso, utilice los [puntos de acceso multirregionales de S3](#), que aceleran las solicitudes a S3 a través de la red global mediante Global Accelerator.

Para obtener más información, consulta [Cómo mejorar la experiencia del jugador aprovechando AWS Global Accelerator y Amazon GameLift FleetIQ](#).

Pasos para la implementación

- Utilice AWS Global Accelerator para mejorar el rendimiento de la red en lo que respecta al tráfico de juegos, el chat de voz y la mensajería en tiempo real, además de facilitar la rápida conmutación por error a los servidores del juego.
- Configura los aceleradores de enrutamiento personalizados de Global Accelerator para integrarlos con tu servicio de búsqueda de parejas, lo que permitirá enrutar de manera determinista a los jugadores a las sesiones de juego mediante la transmisión automática estática. IPs
- Habilite la replicación multiregional de S3 para replicar el contenido compartido en todas las regiones para los equipos de desarrollo de juegos distribuidos.

- Utilice los puntos de acceso multirregionales de S3 para acelerar el acceso a los datos de S3 a través de la red AWS global para los usuarios distribuidos por todo el mundo.

Proceso y cultura

GAMEPERF08: ¿Cómo ajustas el nivel de rendimiento de tu juego a las expectativas de los jugadores y desarrolladores?

Conocer al jugador y al desarrollador es uno de los aspectos más importantes para mejorar la eficiencia del rendimiento. Ofrecer un juego de alto rendimiento con una baja sobrecarga operativa es una de las mejores maneras de demostrar a los jugadores y desarrolladores que te importa su experiencia y que puedes diferenciar tu juego y tu estudio.

Prácticas recomendadas

- [GAMEPERF08-BP01 Informa e incluye al jugador en tu proceso](#)
- [GAMEPERF08-BP02 Alinee la selección de soluciones con las habilidades y la experiencia del equipo de ingeniería](#)

GAMEPERF08-BP01 Informa e incluye al jugador en tu proceso

Ofrece una opción para mostrar en el juego métricas como la latencia, los fotogramas por segundo y los paquetes descartados. Descubre los problemas de infraestructura y los tiempos de inactividad por mantenimiento a través de comunicaciones entre jugadores, como páginas de estado. Celebre las nuevas ubicaciones de los juegos con comunicados de los jugadores, incluidos blogs de desarrolladores, y establezca expectativas sobre las mejoras esperadas en la experiencia de los jugadores.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Incluye al jugador

Proporciona un proceso de envío de diagnósticos sencillo que recopile los archivos relevantes y los adjunte a un ticket de asistencia al jugador desde el cliente del juego. Crea un foro de asistencia en el que los jugadores puedan ayudarse unos a otros y contribuir a mejorar la experiencia de juego

Considera las ventajas y desventajas frente a las expectativas de los jugadores

Es posible que los jugadores no se den cuenta de cambiar los sistemas de back-end por motivos de rentabilidad, pero mover los servidores de juegos puede cambiar el tiempo de ping. Sé coherente y justo con los jugadores a la hora de pensar en ampliar o reducir las ubicaciones de alojamiento de tus juegos.

Las comunidades de jugadores y las zonas geográficas tendrán sus propias características que pueden afectar a las expectativas de tu juego. Por ejemplo, Corea del Sur tiene una de las conexiones a Internet más rápidas del planeta y la expectativa en cuanto a la jugabilidad es una latencia de un solo dígito, lo que impulsa un juego altamente competitivo. La jugabilidad casual en dispositivos móviles crea un perfil de rendimiento y un patrón de uso diferentes a los de la consola y el PC.

El inicio de sesión y el lobby son parte de la experiencia y deberían ser adaptables, incluso si el servidor está desconectado por motivos de mantenimiento. Planificar una noche de redadas o pasar el rato en el lobby forma parte de la experiencia del jugador y es importante tenerlas en cuenta a la hora de elegir áreas prioritarias para mejorar la eficiencia del rendimiento. Los jugadores pueden dejar el cliente del juego abierto durante meses y, a veces, pueden iniciar sesión de vez en cuando para leer las notas del parche. Un juego de Live Ops debe tener en cuenta toda la experiencia del jugador como parte del proceso y la cultura de la ingeniería.

Pasos para la implementación

- Proporciona métricas del juego, como la latencia, los FPS y la pérdida de paquetes, y comunica los problemas de infraestructura y los programas de mantenimiento mediante páginas de estado y actualizaciones para los jugadores.
- Implementa una función de volcado y envío de diagnósticos en el cliente del juego y crea un foro de asistencia para fomentar la resolución de problemas y las mejoras impulsadas por la comunidad.
- Adapta las optimizaciones de rendimiento a las expectativas de la comunidad de jugadores, como una baja latencia para las regiones competitivas o login/lobby experiencias adaptables para los jugadores ocasionales y con sesiones prolongadas.
- Diseña los flujos de trabajo de Live Ops para que tengan en cuenta toda la experiencia del jugador, desde el modo de juego activo hasta el comportamiento de los clientes inactivos, lo que facilitará una interacción fluida.

GAMEPERF08-BP02 Alinee la selección de soluciones con las habilidades y la experiencia del equipo de ingeniería

Evalúe las habilidades y la experiencia de su equipo a la hora de gestionar y optimizar el rendimiento de los servidores de juegos al elegir su opción de alojamiento. Las soluciones autohospedadas, como EC2 los contenedores, requieren más conocimientos sobre la administración de la infraestructura, el ajuste del rendimiento y el escalado. Si tu equipo carece de estas habilidades, un servicio gestionado como este GameLift puede ser más adecuado, ya que elimina muchas de las complejidades y permite a tu equipo centrarse en las optimizaciones específicas del juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Al evaluar estos factores y realizar pruebas de rendimiento en diferentes opciones de alojamiento, puedes seleccionar la solución más adecuada que cumpla con los requisitos específicos de tu juego y, al mismo tiempo, optimice la eficiencia del rendimiento.

Recursos

Obtén más información sobre nuestras prácticas recomendadas relacionadas con la eficiencia del rendimiento.

Documentos relacionados:

- [Centro de arquitectura de AWS](#)
- [El pilar de la eficiencia del rendimiento: AWS un marco de buena arquitectura](#)
- [Comparación de sus patrones de almacenamiento local con los servicios de almacenamiento AWS](#)
- [Almacén de instancias: almacenamiento temporal en bloques para EC2 instancias](#)
- [Recopile métricas, registros y rastreos con el CloudWatch agente](#)
- [CloudWatch¿Agente](#)
- [¿Cómo activo y configuro las redes mejoradas en mis EC2 instancias?](#)
- [Mejora de la experiencia del jugador mediante el uso de AWS Global Accelerator y Amazon FlectiQ GameLift](#)
- [Blog de tecnología de Riot Games: Pruebas de escalabilidad y carga para Valorant](#)
- [Juegos online a gran escala con una solución híbrida AWS](#)

- [Método de saturación y errores de utilización \(USE\)](#)
- [Instancias EC2 puntuales de Amazon](#)
- [Rendimiento a gran escala con Amazon ElastiCache](#)
- [Estrategias de almacenamiento en caché de bases de datos con Redis](#)
- [Opciones de conectividad de Amazon Virtual Private Cloud](#)
- [Patrones de diseño de mejores prácticas: optimización del rendimiento de Amazon S3](#)

Parámetros de referencia relacionados:

- [Compare los volúmenes de Amazon EBS](#)
- [Ancho de banda de red de instancias Amazon EC2](#)

Herramientas relacionadas:

- [Unreal Engine: prueba y optimiza tu contenido](#)
- [Unity Profiler](#)
- [Sistema de calidad Open 3D Engine \(O3DE\)](#)
- [Supervisión de los GameLift servidores Amazon](#)
- [Kit de herramientas GameLift de pruebas de Amazon](#)

Videos relacionados:

- [AWS re:Invent 2019: \[REPEAT 2\] EC2 Fundamentos de Amazon \(CMP211-R2\)](#)
- [AWS re:Invent 2019: Impulsando la próxima generación de Amazon EC2: sumérgase en el sistema Nitro \(03-R2\) CMP3](#)
- [Cómo empezar a usar Amazon GameLift FleetiQ: charlas técnicas AWS en línea](#)
- [Zach Blitz, de Riot Games, habla sobre el uso para mejorar los juegos AWS](#)
- [AWS re:Invent 2023 - AWS Graviton: la mejor relación precio-rendimiento para tus cargas de trabajo \(\) AWS CMP334](#)

Formación relacionada:

- [Alojamiento de servidores de juegos en AWS](#)

- [Uso de Amazon GameLiftFleet IQ para servidores de juegos](#)
- [Introducción a AWS los juegos: parte I](#)
- [Alojamiento de servidores de juegos en AWS](#)
- [AWS re:Invent 2023 — AWS Graviton: la mejor relación precio-rendimiento para sus AWS cargas de trabajo \(\) CMP334](#)

Optimización de costes

El pilar de la optimización de costes incluye el proceso continuo de perfeccionamiento y mejora de un sistema durante todo su ciclo de vida. Este proceso abarca desde el diseño inicial de la primera prueba de concepto hasta el funcionamiento continuo de las cargas de trabajo de producción. Al adoptar las prácticas descritas en este paper, puede crear y operar sistemas rentables que logren los resultados empresariales deseados al precio más bajo. La implementación de estas prácticas de optimización de costos puede permitir a su empresa maximizar el valor de su inversión en la nube.

Los juegos son proyectos creativos únicos que deben competir por la atención y el tiempo de juego de los jugadores. Antes del lanzamiento, los desarrolladores de juegos suelen carecer de una idea clara de cuán popular o duradero será su juego. Según la estrategia de monetización del juego, las prioridades empresariales y la fase del ciclo de vida, los desarrolladores deberán hacer concesiones a la hora de evaluar las decisiones de optimización de costes.

Por ejemplo, durante la fase previa al lanzamiento de un nuevo juego muy esperado, la atención suele centrarse en el desarrollo de las funciones speed-to-market y el rendimiento. La prioridad es comprobar que la infraestructura pueda ampliarse para satisfacer los picos de demanda de jugadores. Por el contrario, si un juego no tiene éxito o su desarrollo se está ralentizando, es posible que el enfoque se centre en reducir los costes en la medida de lo posible para que los jugadores actuales puedan seguir utilizando el juego.

Muchos desarrolladores de juegos también utilizan varios juegos simultáneamente, lo que requiere consideraciones adicionales. Los recursos, como la infraestructura, el software y el personal, pueden compartirse en varios juegos en vivo, lo que permite compensar las pérdidas de un juego con las ganancias de otro. En este escenario, centrarse en la optimización de los costes puede mejorar las finanzas de toda la cartera de juegos.

Dados los modelos de negocio únicos, la escala y la imprevisibilidad de los juegos, las siguientes preguntas clave pueden guiar las decisiones de optimización de costes:

- ¿Cómo puedo medir el coste de la infraestructura por jugador, sistema y función del juego?
- ¿Cuál es el equilibrio adecuado entre la optimización de costes y la experiencia del jugador en la fase actual del ciclo de vida de mi juego?
- ¿Cómo puedo maximizar el retorno de la inversión utilizando el modelo de precios adecuado para mis AWS recursos?

Aplicar estas prácticas recomendadas y formular las preguntas correctas puede ayudar a los desarrolladores de juegos a crear y operar sistemas rentables que logren resultados empresariales y, al mismo tiempo, minimicen los costos.

Áreas de interés

- [Principios de diseño](#)
- [Práctica de administración financiera en la nube](#)
- [Conocimiento del gasto y del uso](#)
- [Recursos rentables](#)
- [Costos de transferencia de datos](#)
- [Administrar los recursos de oferta y demanda](#)
- [Optimización a lo largo del tiempo](#)
- [Recursos](#)

Principios de diseño

Además de los principios de diseño del pilar de optimización de costes del Well-Architected Framework, los siguientes principios de diseño optimizan los costes de ejecutar la carga de trabajo de los juegos en la nube.

- Mida el coste de la infraestructura por jugador, sistema y función del juego: comprenda y realice un seguimiento de los costes de infraestructura necesarios para ofrecer experiencias y funciones específicas de los jugadores en todos los sistemas de juego. Esto puede identificar las áreas de su arquitectura que pueden necesitar una optimización de costos.
- Evalúa la relación entre la optimización de costes y la experiencia del jugador: evalúa en qué fase se encuentra tu juego para determinar cuál es el enfoque correcto: la experiencia del jugador o la optimización de costes. Por lo general, una vez que un juego alcanza una masa crítica y la población de jugadores se estabiliza, es hora de centrarse en optimizar los costes operativos. La clave es encontrar un equilibrio entre ofrecer una gran experiencia al jugador y gestionar la infraestructura del juego de la manera más rentable. La implementación de estos principios de diseño maximiza el retorno de tus inversiones en juegos.

Práctica de administración financiera en la nube

No existen prácticas recomendadas de gestión financiera en la nube específicas para Games Lens. Consulte el pilar de [optimización de costes de Well-Architected Framework](#) para obtener orientación sobre la gestión financiera en la nube.

Conocimiento del gasto y del uso

GAMECOST01: ¿Cómo se mide el coste de los entornos de juego?

Comprende el coste por jugador, las características del juego y el entorno para poder gestionar y pronosticar tus gastos a medida que el número de jugadores cambia con el tiempo y se añaden y mejoran funciones. Ten en cuenta las siguientes prácticas recomendadas para gestionar los costes de los distintos entornos de juego.

Prácticas recomendadas

- [GAMECOST01-BP01 Implementa la atribución del coste por jugador, función del juego y entorno](#)
- [GAMECOST01-BP02 Descubre oportunidades de optimización](#)

GAMECOST01-BP01 Implementa la atribución del coste por jugador, función del juego y entorno

La atribución de costes para los servidores de juegos suele ser más sencilla de realizar que los servicios de backend de juegos, ya que un servidor de juegos suele estar optimizado para poder alojar un número específico de jugadores simultáneos por instancia, lo que se puede amortizar con el coste de funcionamiento de la instancia.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

En el caso de los servicios de backend de juegos, se recomienda desvincular los componentes del juego en funciones distintas que se puedan gestionar como recursos lógicos o físicos independientes para facilitar el análisis de los costes.

Por ejemplo, si bien puede parecer sencillo implementar una sola aplicación monolítica para alojar los servicios de back-end de los juegos, este patrón hace que sea difícil calcular el costo total por jugador y función del juego a lo largo del tiempo a medida que se agregan más funciones, ya que los costos de procesamiento, redes y almacenamiento de los recursos se comparten entre las funciones. Considere la posibilidad de adoptar una arquitectura sin servidor para los servicios de backend de sus juegos con servicios como [Amazon API Gateway](#) AWS Lambda y/o AWS Fargate para computación, [Amazon SQS y Amazon SNS para mensajería](#), [Amazon S3](#) para almacenamiento de objetos y Amazon DynamoDB para almacenamiento de bases de datos. Estos servicios son solo algunos ejemplos de productos que ofrecen precios basados en el uso y basados principalmente en el volumen de solicitudes, de modo que los costos se puedan visualizar con detalle. Los recursos individuales, como las funciones de Lambda, los servicios de Fargate, las tablas de DynamoDB y los depósitos de S3, se pueden asociar a las etiquetas de asignación de costes para que pueda atribuir los costes de estos servicios a nombres de funciones de los juegos que le faciliten la comprensión de los costes de cada uno de sus servicios.

También se recomienda administrar por separado cada uno de los entornos de desarrollo de juegos para poder atribuir los costos de los distintos entornos. Por lo general, los desarrolladores de juegos gestionan entornos separados para los entornos de desarrollo, prueba, puesta en escena y producción, tal y como se describe en el pilar de operaciones de esta industria de los videojuegos. Por lo general, cada entorno tiene requisitos de escalabilidad, rendimiento y uso diferentes y puede ser gestionado por equipos independientes. Para controlar los costos, organice estos entornos de manera que pueda monitorear y atribuir adecuadamente los costos de cada entorno.

Para obtener más información, consulte la siguiente documentación:

- [Creación de un juego multijugador sin servidor que escale](#)
- [Servidores de sesión de juego independientes con un backend basado WebSockets](#)
- [Servidores de sesión de juego independientes con un backend sin servidor](#)

Pasos para la implementación

- Separe los servicios de backend de los juegos en distintas funciones mediante arquitecturas sin servidor o en contenedores, como Amazon API AWS Lambda Gateway, y permita una atribución granular de los costes por AWS Fargate función.
- Aplique etiquetas de asignación de costos a recursos individuales (por ejemplo, funciones Lambda, tablas de DynamoDB y depósitos de S3) para asociar los costos con características específicas del juego y así poder analizar mejor los costos.

- Gestione entornos independientes para el desarrollo, las pruebas, la puesta en escena y la producción, y organice y supervise sus costes de forma independiente para adaptarlos a los requisitos de escalabilidad y uso.

GAMECOST01-BP02 Descubre oportunidades de optimización

Los desarrolladores y editores de juegos pueden utilizar AWS FinOps prácticas que les ayuden a optimizar sus costes en la nube y a obtener una mejor visibilidad de sus gastos en la nube. De este modo, los productores de juegos pueden alinear el coste medio necesario para mantener la infraestructura para los jugadores con los resultados financieros que genera el juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: bajo

Guía para la implementación

AWS ofrece una [guía de soluciones listas para usar para la gestión financiera en la nube](#) con el fin de gestionar y optimizar los gastos de los servicios en la nube. Esta capacidad incluye una visibilidad pormenorizada y un análisis de costes y uso para facilitar la toma de decisiones sobre temas como los cuadros de mando de gastos, la optimización, los límites de gasto, la devolución de gastos y la detección y respuesta a las anomalías. La guía de soluciones para la gestión financiera en la nube incluye funciones de presupuesto y previsión, lo que le proporciona una arquitectura definida y optimizada en cuanto a costes para sus cargas de trabajo, de modo que pueda seleccionar el modelo de precios adecuado y atribuir los costes de los recursos a sus equipos. Esto activa las técnicas de seguimiento, notificación y optimización de costes en todo su entorno y sus recursos. Puede gestionar de forma centralizada la información sobre los gastos y dar acceso a las partes interesadas fundamentales según sea necesario para obtener una visibilidad específica y respaldar la toma de decisiones.

Otra FinOps herramienta clave es el [centro de optimización de costes](#), que ofrece una visión centralizada de las recomendaciones y oportunidades de optimización de costes en todas sus Cuentas de AWS instalaciones Regiones de AWS, para que pueda aprovechar al máximo sus AWS gastos. Puedes usar Cost Optimization Hub para identificar, filtrar y agregar recomendaciones de optimización de AWS costos en todas tus Cuentas de AWS instalaciones Regiones de AWS. Hace recomendaciones sobre el redimensionamiento de los recursos, la eliminación de los recursos inactivos, los Savings Plans y las instancias reservadas. Con un único panel, se evita tener que ir a varios AWS productos para identificar las oportunidades de optimización de costes.

Si sus equipos de juegos utilizan MyApplications de forma compartida Cuentas de AWS , puede utilizar [MyApplications in Consola de administración de AWS Home](#) para ver los costes de los recursos de las aplicaciones correspondientes a cargas de trabajo individuales. Esta vista detallada te permite identificar las tendencias de costes específicas de tu infraestructura de juegos, lo que te permite tomar decisiones informadas sobre la asignación y optimización de los recursos.

Además, al revisar periódicamente los datos de facturación y gestión de costes con [AWS Data Exports](#), se descubren oportunidades ocultas de ahorro de costes. Este informe detallado proporciona un desglose completo de sus gastos en la nube, lo que le permite identificar las áreas de gasto excesivo, los recursos no utilizados y las oportunidades para aprovechar servicios o modelos de precios más rentables.

Al adoptar FinOps los principios y aprovechar las herramientas que ofrecen AWS, los desarrolladores y editores de juegos pueden hacer un uso más eficiente de sus recursos en la nube y, en última instancia, mejorar sus resultados y liberar fondos para seguir desarrollando e innovando en los juegos.

Pasos para la implementación

- Utilice las herramientas de gestión Nube de AWS financiera para obtener una visibilidad pormenorizada y detallada, paneles de gastos, detección de anomalías y atribución de costes para optimizar y realizar un seguimiento eficaz de los gastos en la nube.
- Utilice el centro de optimización de costes para centralizar las recomendaciones de Rightsizing, Savings Plans e Reserved Instance en Cuentas de AWS todas las regiones.
- Revise periódicamente los datos AWS de facturación mediante Data Exports y, así, MyApplication podrá analizar AWS los costes específicos de la carga de trabajo, descubrir oportunidades de ahorro y optimizar la asignación de recursos.

Recursos rentables

GAMECOST02: ¿Cómo eliges la solución informática adecuada para tus servidores de juegos?

Uno de los aspectos más exclusivos de la carga de trabajo de un juego, en comparación con otros tipos de cargas de trabajo, es el servidor del juego. El servidor del juego es fundamental para la

experiencia del jugador, ya que los jugadores se conectan a él desde su cliente de juego para jugar una sesión de juego.

El servidor de juegos también es uno de los principales impulsores del coste de funcionamiento de un juego multijugador. Por lo tanto, es importante optimizar la forma en que se utiliza la infraestructura informática de los servidores de juegos para reducir los costes.

Prácticas recomendadas

- [GAMECOST02-BP01 Optimiza el coste de la transferencia de datos a través de Internet](#)
- [GAMECOST02-BP02 Optimiza el número de sesiones de juego alojadas en cada instancia del servidor de juegos para optimizar los costes](#)
- [GAMECOST02-BP03 Seleccione la opción de precio de computación adecuada para reducir los costos](#)

GAMECOST02-BP01 Optimiza el coste de la transferencia de datos a través de Internet

Si bien la transferencia de datos salientes (egresados) de tus AWS recursos a Internet se centra en AWS principalmente en los costes de transferencia de datos, las empresas de videojuegos se enfrentan a costes elevados relacionados con la transferencia de datos a través de AWS Direct Connect de los balanceadores de carga AWS Gateway, que pueden cobrar tanto por los datos entrantes (entrantes) como por los salientes. Implemente soluciones que reduzcan el costo total de transferir datos desde el AWS backend del juego a sus jugadores, centrándose en minimizar los cargos de salida de sus AWS recursos y evaluando las opciones para administrar las tarifas de entrada y salida a través de los servicios de AWS conectividad.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Utilice Amazon CloudFront para reducir el coste de la entrega de contenido y de las aplicaciones web orientadas al público.

El contenido y los activos del juego que se almacenan en la nube suelen almacenarse en Amazon S3 y entregarse al cliente del juego directamente desde S3 o desde servidores web alojados en Amazon EC2 que recuperan el contenido de Amazon S3 y lo entregan a los clientes. Para reducir los costos de transferencia de datos de las descargas de contenido, considere usar Amazon antes que su almacenamiento CloudFront en la nube para entregar contenido a los usuarios.

El uso CloudFront puede reducir el costo de la transferencia de datos, ya que cuesta menos entregar el contenido desde las regiones CloudFront points-of-presence que directamente desde ellas y CloudFront no se cobran tarifas de recuperación de origen para los orígenes AWS basados, como Amazon EC2 y Amazon S3. Si su contenido es estático y no cambia con frecuencia, puede usarlo CloudFront para almacenar esos datos en caché más cerca de los usuarios finales, lo que puede reducir aún más los costos.

CloudFront también mejora la rentabilidad de las aplicaciones y servicios web orientados al público desde el principio, incluso si no se utiliza el almacenamiento en caché, ya que el coste de la transferencia de datos entre los servidores y los clientes se puede reducir al enrutar el tráfico a través de la red. AWS

CloudWatchPuedes usar [Amazon](#) para monitorear tu CloudFront uso de Amazon. Para los casos de uso en los que utilice varias redes de entrega de contenido (CDNs), [Amazon CloudFront Origin Shield](#) puede proporcionar una capa adicional de almacenamiento en caché para consolidar y reducir el número de solicitudes de origen de distintos proveedores.

Para comprender el tráfico de la red del juego, puede habilitar los [registros de flujo de VPC](#) y [Amazon CloudWatch Internet Monitor](#) para que end-to-end puedan ver las conexiones del backend del jugador o del juego. Este enfoque permite identificar las causas del alto costo de transferencia de datos y realizar cambios en la arquitectura para optimizar el gasto en transferencia de datos.

Pasos para la implementación

- Utilice Amazon CloudFront frente a Amazon S3 o a los orígenes de contenido EC2 basados en Amazon para reducir los costes de transferencia de datos al aprovechar los costes de envío desde CloudFront points-of-presence y eliminar las tarifas de recuperación de origen.
- Habilite los registros de flujo de VPC y Amazon CloudWatch Internet Monitor para analizar el tráfico de la red e identificar los cambios en la arquitectura para optimizar los costos de transferencia de datos.
- Implemente CloudFront Origin Shield para consolidar y reducir las solicitudes de origen cuando utilice varias solicitudes CDNs para aumentar la rentabilidad.

Para obtener más información sobre las mejores prácticas de entrega de contenido, consulta el [documento técnico sobre entrega de contenido para juegos](#).

GAMECOST02-BP02 Optimiza el número de sesiones de juego alojadas en cada instancia del servidor de juegos para optimizar los costes

Optimice la cantidad de sesiones de juego alojadas por instancia de servidor para lograr una mejor utilización de la computación y reducir los costos de la infraestructura de computación.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Para optimizar los costes, los desarrolladores de juegos deben maximizar el número de sesiones de juego alojadas en el mismo servidor físico o virtual, lo que también se conoce como densidad de empaquetado de sus servidores de juegos. Esto se consigue aumentando el número de procesos del servidor de juegos que se pueden alojar simultáneamente en una instancia.

Por lo general, un único proceso de servidor de juegos no debería requerir el uso de todos los recursos disponibles en la EC2 instancia. Esta es una de las formas más importantes de reducir los costes informáticos de un juego y requiere el uso de un software que pueda generar y gestionar varios procesos de servidor en la EC2 instancia en puertos separados.

Por ejemplo, Amazon GameLift tiene una cuota en el número máximo de procesos de servidor de juegos por instancia, que debes esforzarte por utilizar para reducir los costes de alojamiento. Para obtener más información, consulta los [puntos finales y las cuotas de Amazon GameLift Servers](#) para obtener más información sobre la cuota actual del máximo de procesos de servidor de juegos por instancia.

Como alternativa a la implementación de procesos de servidores de juegos en máquinas virtuales, como las EC2 instancias, cada vez es más popular entre los desarrolladores de juegos ejecutar sus servidores de juegos como aplicaciones basadas en contenedores utilizando soluciones de orquestación de contenedores. Los desarrolladores de juegos pueden utilizar [Amazon Elastic Container Service](#) (Amazon ECS) o [Guidance for Game Server Hosting con Agones y Open Match en Amazon EKS](#). Otra opción es [Game Server Hosting on AWS Fargate](#), un motor informático sin servidor que funciona tanto con ECS como con EKS, lo que le permite centrarse en el juego sin tener que gestionar la infraestructura subyacente.

Las soluciones de contenedores proporcionan una funcionalidad de programación de tareas que permite encontrar automáticamente una instancia de contenedor disponible en el clúster para alojar el contenedor del servidor de juegos en función de los requisitos de recursos y de cualquier otra lógica de ubicación que especifique. Sin embargo, es importante tener en cuenta cómo gestionarás

el comportamiento de escalado y ubicación de los jugadores de forma que no interrumpen las sesiones activas de los jugadores.

Pasos para la implementación

- Aumente la densidad de empaquetado ejecutando varios procesos de servidor de juegos por EC2 instancia utilizando puertos independientes y software de administración de procesos.
- Usa Amazon GameLift o soluciones de contenedores como ECS o EKS AWS Fargate para gestionar los procesos de los servidores de juegos de forma eficiente y reducir los costes de infraestructura.
- Supervisa continuamente el uso de los recursos para refinar la densidad de empaque y mantener la rentabilidad sin comprometer la experiencia del jugador.

GAMECOST02-BP03 Seleccione la opción de precio de computación adecuada para reducir los costos

Realiza pruebas de rendimiento del software de tu servidor de juegos en una variedad de tipos de instancias y opciones de procesamiento para determinar cuál es la opción más rentable para tu juego.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Además de utilizar de forma eficiente los tipos de EC2 instancias adecuados para su carga de trabajo, considere cuál de las opciones de precios de procesamiento disponibles es la más adecuada para sus objetivos de optimización de costes. Hay varias opciones de precios disponibles, incluidas On-Demand Instances, Spot Instances, Reserved Instances y Savings Plans.

[Los Savings Plans](#) (SPs) ofrecen descuentos en informática mediante compromisos de uso y son ideales para situaciones en las que no se puede pronosticar el uso previsto para un período de 1 o 3 años. Ofrecen descuentos, como las instancias reservadas, con la flexibilidad de aplicarlos en todas las regiones, familias de instancias, sistemas operativos y arrendamientos. También se pueden aplicar a AWS Fargate quién puede ser una opción de alojamiento de servidores de juegos para juegos casuales o AWS Lambda quién se utiliza como una buena opción para juegos por turnos que no requieren servidores de juegos. Para obtener más información, consulta Cómo [crear un juego multijugador sin servidor que sea](#) escalable.

Los Savings Plans se introducen durante el lanzamiento de los juegos para ahorrar costes en las cargas de trabajo de los servidores de juegos que contribuyen a la inversión de EC2 instancias cuando el juego se lanza al público. Los Savings Plans también se pueden introducir después del lanzamiento, cuando el equipo de operaciones del juego comprenda mejor el tráfico de jugadores después de que el juego haya estado en producción durante un período prolongado.

Dado que los Savings Plans ofrecen flexibilidad regional, son especialmente ideales para optimizar el gasto en servidores de juegos para juegos con un uso impredecible en todas las geografías.

Por ejemplo, si su patrón de uso diario de jugadores requiere al menos 20 servidores para mantener su base de jugadores, pero periódicamente requiere hasta 40 servidores, considere la posibilidad de adquirir compromisos del Plan de Ahorros para cubrir la base de 20 servidores, ya que esa demanda de uso es predecible y constante, y se traducirá en el máximo aprovechamiento del compromiso de uso que ha adquirido.

Maximice el uso de los Savings Plans y amplíelos con otras opciones de compra que ofrezcan más flexibilidad ante picos impredecibles de uso de los servidores de juegos, como las instancias bajo demanda y las instancias puntuales, para lograr ahorros óptimos.

Las instancias puntuales son ideales para ejecutar servidores de juegos porque ofrecen los mayores descuentos en informática, no requieren compromisos de uso y proporcionan flexibilidad para tipos de cargas de trabajo impredecibles y con picos de trabajo. Sin embargo, las instancias puntuales se pueden interrumpir, por lo que son más adecuadas para cargas de trabajo de servidores de juegos con sesiones de juego cortas o situaciones en las que la tolerancia a la interrupción es mayor.

Para obtener más información sobre las instrucciones para ejecutar servidores de juegos mediante Kubernetes en Amazon EKS con instancias EC2 puntuales, consulte [Cómo ejecutar juegos multijugador masivos con Spot EC2 mediante Aurora Serverless](#).

Utilice [Amazon EC2 Spot Instances](#) para determinar los grupos con la menor probabilidad de interrupción que generarán el máximo ahorro en comparación con las tarifas bajo demanda.

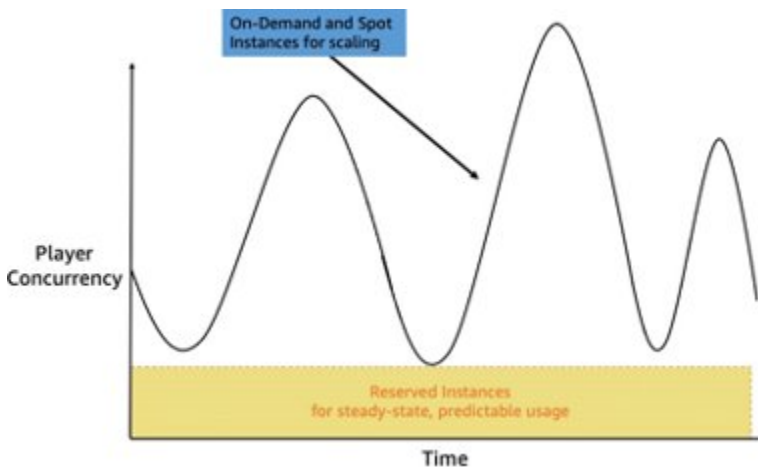
Al utilizar Spot, también se recomienda ejecutar las cargas de trabajo de los servidores de juegos en varios tipos de EC2 instancias y zonas de disponibilidad Región de AWS para diversificar el uso de la capacidad y reducir el riesgo de interrupciones.

Considere la posibilidad de utilizar instancias puntuales en combinación con instancias bajo demanda para minimizar el impacto de las posibles interrupciones en las sesiones de juego activas y utilice una estrategia de asignación de capacidad optimizada para reducir aún más el riesgo de interrupciones.

Consulta las [prácticas recomendadas de Amazon EC2 Spot para](#) obtener más información sobre las prácticas recomendadas. El [reequilibrio de capacidad en Auto Scaling para reemplazar las instancias puntuales en riesgo](#) se puede utilizar para monitorear proactivamente y agregar capacidad adicional cuando las instancias puntuales corren un mayor riesgo de interrupción.

[Amazon GameLift FleetIQ](#) se integra con las instancias puntuales para optimizar el uso de instancias puntuales de bajo coste y, al mismo tiempo, reducir el riesgo de interrupciones. Si aloja su juego con recursos informáticos GameLift, consulte la GameLift documentación para elegir los recursos informáticos. Para obtener más información, consulte [Elegir recursos informáticos para una flota gestionada](#).

El siguiente diagrama proporciona un ejemplo para ilustrar el uso de varias opciones de precios de procesamiento para las cargas de trabajo de los servidores de juegos:



Alojamiento de servidores de juegos con múltiples opciones EC2 de precios

En el diagrama, la simultaneidad de los jugadores fluctúa con el tiempo, lo que dificulta la gestión de la utilización y la optimización de los costes. Para hacer frente a esta fluctuación, considere la posibilidad de adoptar una combinación de diferentes opciones de precios informáticos, utilizando Savings Plans EC2 para satisfacer las necesidades de sus requisitos de uso mínimo y, al mismo tiempo, confiar en las instancias EC2 puntuales y EC2 bajo demanda para satisfacer las necesidades de sus jugadores.

Pasos para la implementación

- Utilice los Savings Plans para un uso básico predecible y combínelos con instancias puntuales y bajo demanda para obtener flexibilidad y optimizar los costos durante los picos de uso.

- Utilice instancias puntuales para servidores de juegos con sesiones de corta duración o una mayor tolerancia a las interrupciones, diversificando los tipos de instancias y las zonas de disponibilidad para minimizar el riesgo.
- Implemente herramientas como EC2 Spot Instances Advisor, Capacity Rebalancing y GameLift FleetIQ para optimizar el uso de las instancias puntuales y gestionar las interrupciones de forma proactiva.

Costos de transferencia de datos

GAMECOST03: ¿Cómo optimizas los costes de transferencia de datos para tu infraestructura de juegos?

Los juegos pueden transferir una cantidad significativa de datos a través de Internet entre los dispositivos cliente de juego de tus jugadores y tu infraestructura de juego para proporcionar la experiencia de juego, así como entre los componentes de tu infraestructura de juego.

Por ejemplo, la transferencia de datos se produce cuando los jugadores descargan actualizaciones del contenido del juego en sus clientes de juego, guardan su progreso en la nube, participan en partidas multijugador en tiempo real con sus amigos y cuando la infraestructura del juego transfiere datos entre regiones y zonas de disponibilidad. Es importante saber dónde se produce la transferencia de datos en la carga de trabajo del juego para optimizar las opciones de arquitectura y reducir este coste de transferencia de datos.

Para optimizar los costes de transferencia de datos para la carga de trabajo de tu juego, ten en cuenta las siguientes prácticas recomendadas:

Prácticas recomendadas

- [GAMECOST03-BP01 Elija el tipo de almacenamiento apropiado para el contenido generado por el usuario a fin de reducir los costos](#)
- [GAMECOST03-BP02 Optimiza las bases de datos para los backends de los juegos](#)

GAMECOST03-BP01 Elija el tipo de almacenamiento apropiado para el contenido generado por el usuario a fin de reducir los costos

Cada tipo de datos generados y almacenados en tu juego tiene características únicas que debes tener en cuenta a la hora de determinar la solución de almacenamiento adecuada para tu carga de trabajo.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: bajo

Guía para la implementación

Utilice Amazon S3 Object Lifecycle Management para almacenar datos de objetos en la clase de almacenamiento más rentable. Amazon S3 ofrece varias [clases de almacenamiento](#) y una [gestión del ciclo de vida de los objetos](#) para facilitar la configuración de políticas sencillas y detalladas para la transición automática de los datos entre los niveles de almacenamiento y reducir los costes. En lugar de simplemente almacenar los datos en la clase de almacenamiento estándar S3 de forma predeterminada, considere establecer una configuración de ciclo de vida para que los datos pasen automáticamente de un nivel a otro a lo largo del tiempo, o utilice la clase de almacenamiento S3 Intelligent-Tiering para patrones de acceso desconocidos o cambiantes.

Como alternativa, S3 Intelligent-Tiering puede realizar la transición de datos entre niveles de forma rentable y automática, y se recomienda como clase de almacenamiento predeterminada, ya que optimiza los costos sin necesidad de configurar manualmente las políticas del ciclo de vida, y ahora es la mejor opción para objetos pequeños y de corta duración. Para obtener más información, consulte [Amazon S3 Intelligent-Tiering: optimizaciones de costos mejoradas para objetos pequeños y de corta duración](#).

Los casos de uso más comunes de Amazon S3 incluyen el almacenamiento de activos de juegos, contenido estático, registros de juegos, almacenamiento de lagos de datos y copias de seguridad. Para los casos de uso en los que se requieran sistemas de archivos, como conectar sistemas de archivos compartidos a las estaciones de trabajo durante el desarrollo, considere la posibilidad de utilizar [Amazon Elastic File System \(Amazon EFS\)](#), que proporciona diferentes clases de almacenamiento y aumenta y reduce automáticamente a medida que agrega y elimina archivos sin necesidad de administrar la infraestructura.

[Amazon S3 One Zone -IA](#) es una opción de almacenamiento ideal para datos transitorios relacionados con sesiones de juego, partidas u otra información efímera que se pueda volver a crear según sea necesario. Este tipo de datos del juego no requiere redundancia en varias zonas

de disponibilidad (). AZs Esta clase de almacenamiento de menor coste es ideal para registrar las acciones de los jugadores, los eventos del juego y otros datos de telemetría que se utilizan para realizar análisis o depurar errores.

La principal ventaja de optimización de costes de utilizar S3 Express One Zone para estos datos de juegos es el importante ahorro de costes en comparación con la clase de almacenamiento S3 estándar, con una reducción de hasta un 20% en los costes de almacenamiento. Esto puede resultar especialmente ventajoso para los juegos con grandes volúmenes de datos que no requieren el mismo nivel de durabilidad y disponibilidad que los datos de aplicaciones esenciales. Al aprovechar S3 One Zone, los desarrolladores y editores de juegos pueden optimizar sus costes de almacenamiento en la nube sin comprometer la experiencia general del jugador.

Pasos para la implementación

- Configure las políticas del ciclo de vida de Amazon S3 para realizar la transición de datos entre clases de almacenamiento o utilice S3 Intelligent-Tiering de forma predeterminada para optimizar automáticamente los costes con los cambios en los patrones de acceso.
- Utilice S3 One Zone-Infrequent Access para obtener datos transitorios de las sesiones de juego, como los registros de telemetría y emparejamiento, a fin de reducir los costes de almacenamiento hasta en un 20% y, al mismo tiempo, mantener una disponibilidad suficiente.
- Para las necesidades de sistemas de archivos compartidos durante el desarrollo, utilice Amazon EFS para simplificar la administración del almacenamiento con una capacidad elástica y varias clases de almacenamiento.

GAMECOST03-BP02 Optimiza las bases de datos para los backends de los juegos

Los juegos dependen en gran medida de las bases de datos para almacenar una amplia gama de datos críticos, desde los perfiles e inventarios de los jugadores hasta las microtransacciones del juego y las métricas de progreso. Las bases de datos también desempeñan un papel crucial en la gestión de los aspectos sociales de los juegos, como la creación y el mantenimiento de grupos de jugadores y grupos, y la aplicación de políticas de moderación. A medida que aumente la base de jugadores de un juego, los costes asociados a las bases de datos aumentarán inevitablemente para adaptarse a la creciente demanda de datos y uso.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: medio

Guía para la implementación

Para los backends de juegos que se ejecutan en Amazon Aurora, se pueden emplear varias estrategias de optimización de costos. Una recomendación clave es [escalar automáticamente las réplicas de lectura en función de los patrones de uso](#), aumentando o reduciendo dinámicamente el número de réplicas para gestionar las fluctuaciones del tráfico. Esto significa que está pagando por los recursos que realmente necesita. Otra táctica de optimización consiste en sustituir las réplicas de lectura utilizadas para el análisis de juegos por exportaciones de instantáneas de bases de datos a Amazon S3, ya que el servicio de almacenamiento de S3 suele ser más asequible que las instancias de bases de datos Aurora aprovisionadas. Para obtener más información, consulte [Exportación de datos de instantáneas de bases de datos a Amazon S3 para Amazon RDS](#).

Explorar el uso de [instancias de base de datos reservadas para Amazon Aurora](#) para sus instancias de bases de datos principales y realizar la transición a la configuración [Aurora Serverless](#) también puede generar importantes ahorros de costos a largo plazo, ya que proporciona más flexibilidad y un [control granular de la utilización de los recursos](#).

Del mismo modo, para los backends de juegos que utilizan Amazon DynamoDB, utilizar [el modo de capacidad bajo demanda de DynamoDB](#) puede ser una opción eficaz, especialmente para cargas de trabajo nuevas o impredecibles, ya que le permite pagar solo por los recursos que consume sin necesidad de aprovisionar en exceso. A medida que los patrones de tráfico de los juegos se vayan haciendo más estables y predecibles con el paso del tiempo, podrá pasar al modo de capacidad [aprovisionada de DynamoDB, que le permitirá ahorrar costes gracias a una mejor planificación de la capacidad](#). La activación del autoscaling en las tablas de DynamoDB es otra optimización clave, ya que permite al servicio ajustar dinámicamente la capacidad aprovisionada en función de las fluctuaciones del tráfico. Pruebe la estructura de datos del juego en un entorno de desarrollo antes del lanzamiento para encontrar y eliminar los [índices secundarios locales \(\)](#) y los [índices secundarios globales \(LSIs\)](#) innecesarios. GSIs Esto puede suponer un importante ahorro de costes en el almacenamiento y las operaciones de los datos del juego. Eliminar [las operaciones de escaneo ineficientes](#) del código de backend del juego en favor de consultas más específicas, comprar la capacidad [reservada de Amazon DynamoDB y aprovechar DynamoDB Streams AWS Lambda con activadores para procesar los eventos del backend del juego puede optimizar aún más sus costos de DynamoDB](#). Para obtener más información, consulte [Prácticas recomendadas para consultar y analizar datos en DynamoDB](#).

Al implementar estas estrategias de optimización de costos tanto para Amazon Aurora como para DynamoDB, los desarrolladores y editores de juegos pueden reducir considerablemente sus gastos en bases de datos de back-end de juegos.

Pasos para la implementación

- Utilice el autoscalamiento de réplicas de lectura y las exportaciones de instantáneas de bases de datos de Aurora a Amazon S3 para gestionar de forma rentable las fluctuantes necesidades de tráfico y análisis.
- Optimice los costos de DynamoDB empezando por la capacidad bajo demanda para nuevas cargas de trabajo, pasando a la capacidad aprovisionada con autoscalamiento para un tráfico predecible y eliminando las no utilizadas y. LSIs GSIs
- Evite operaciones de escaneo ineficientes en favor de consultas específicas, utilice instancias reservadas o capacidad reservada y utilice DynamoDB Streams para el procesamiento de eventos.
AWS Lambda

Administrar los recursos de oferta y demanda

No existen mejores prácticas de gestión de los recursos de oferta y demanda específicas para Games Lens.

Para obtener más información sobre la gestión de la demanda y el suministro de recursos, consulte [Cost Optimization Pillar: AWS Well-Architected Framework](#).

Optimización a lo largo del tiempo

No existen mejores prácticas de optimización a lo largo del tiempo específicas de Games Lens.

Para obtener más información sobre cómo optimizar los costos a lo largo del tiempo, consulte el [pilar de la optimización de costos: AWS Well-Architected Framework](#).

Recursos

Consulte los siguientes recursos para obtener más información sobre las mejores prácticas para la optimización de costos:

Documentos relacionados:

- [¿Cómo puedo reducir los cargos por transferencia de datos de mi puerta de enlace NAT en Amazon VPC?](#)
- [Presentamos el adaptador Amazon GameLift FleetIQ para Agones](#)

- [¿Cómo puedo encontrar los principales contribuyentes al tráfico de la puerta de enlace NAT en mi Amazon VPC?](#)
- [Elija la estrategia de cómputo adecuada para sus servidores de juegos globales](#)
- [AWS Well-Architected Labs: recursos rentables](#)
- [El complemento CNI de Amazon VPC aumenta los límites de pods por nodo](#)
- [Mejores prácticas de arquitectura para la optimización de costos](#)
- [Reducir el tiempo de espera de los jugadores y ajustar el tamaño correcto de la asignación de cómputo con Amazon SageMaker AI RL y Amazon EKS](#)
- [AWS Compute Optimizer](#)
- [Electronic Arts optimiza los costes y las operaciones de almacenamiento con Amazon S3 Intelligent-Tiering y Amazon Glacier](#)
- [Evite las prácticas de licenciamiento poco amigables migrando las cargas de trabajo de Windows a Linux](#)
- [Overview of Data Transfer Costs for Common Architectures](#)
- [AWS y Kubecost colaboran para ofrecer un monitoreo de costos a los clientes de EKS](#)
- [Sentar las bases: configurar su entorno para la optimización de costes](#)
- [Descripción general de Amazon EC2 Spot Instances](#)

Sostenibilidad

El pilar de sostenibilidad proporciona principios de diseño, orientación operativa, mejores prácticas y planes de mejora para ayudar a cumplir los objetivos de sostenibilidad de sus cargas de AWS trabajo.

Puede encontrar una guía de implementación sobre la implementación en el documento técnico sobre el [pilar de la sostenibilidad: el AWS marco de una buena arquitectura](#).

Áreas de interés

- [Principios de diseño](#)
- [Selección de región](#)
- [Alineación con la demanda](#)
- [Software y arquitectura](#)
- [Administración de datos](#)
- [Hardware y servicios](#)
- [Recursos](#)

Principios de diseño

La sostenibilidad de la industria de los videojuegos está evolucionando a ritmos diversos en las distintas regiones del mundo. Con la energía sostenible en grandes franjas de América del Norte y los mandatos de sostenibilidad en la UE y el Reino Unido, los arquitectos tienen varios enfoques para lograr cargas de trabajo más ecológicas en un futuro próximo. El pilar de [sostenibilidad de Well-Architected se](#) puede utilizar para lograr estas medidas para una amplia variedad de cargas de trabajo.

En esta sección del objetivo se describen varias prácticas recomendadas que se pueden utilizar para las cargas de trabajo de los juegos.

- Seleccione el tipo de almacenamiento adecuado para los datos de usuario de los juegos.
- Tenga en cuenta las políticas del ciclo de vida de los datos que eliminarán la duplicación de los datos y eliminarán los datos innecesarios de sus cargas de trabajo.
- Sea selectivo a la hora de dimensionar correctamente la implementación de los recursos informáticos.

- Utilice la tecnología sin servidor para procesos breves y transaccionales.

Selección de región

No hay prácticas recomendadas de [selección de regiones](#) específicas para Games Lens. Para obtener más información, consulte el [pilar de sostenibilidad: AWS Well-Architected Framework](#).

Alineación con la demanda

No existe ningún acuerdo [que exija](#) las mejores prácticas específicas desde el punto de vista de los juegos. Para obtener más información, consulte el [pilar de sostenibilidad: AWS Well-Architected Framework](#).

Software y arquitectura

No hay prácticas recomendadas de [software y arquitectura](#) específicas para Games Lens. Para obtener más información, consulte el [pilar de sostenibilidad: AWS Well-Architected Framework](#).

Administración de datos

GAMESUS01: ¿Cómo gestionas los datos de usuario y juego en tu sistema de juego?

Desarrolle una estrategia para el ciclo de vida de los datos que optimice el almacenamiento y la relevancia de los datos conservando únicamente los datos históricos críticos.

Prácticas recomendadas

- [GAMESUS01-BP01 Usa tecnologías de almacenamiento que se adapten a los patrones adaptados al contenido del usuario, la información de los suscriptores y las compras dentro del juego](#)
- [GAMESUS01-BP02 Usa políticas de ciclo de vida o caducidad de TTL para eliminar juegos innecesarios, datos de usuario, archivos de registro o activos obsoletos.](#)

GAMESUS01-BP01 Usa tecnologías de almacenamiento que se adapten a los patrones adaptados al contenido del usuario, la información de los suscriptores y las compras dentro del juego

Debe clasificar sus datos por tipo, necesidad de retención y frecuencia de acceso. Esto le permite seleccionar la solución de almacenamiento más optimizada para los innumerables tipos de datos que generan sus juegos o servicios de backend. Los datos que cambian rápidamente deben almacenarse en servicios de bases de datos con valores clave o en memoria. Los datos transaccionales deben almacenarse en servicios de bases de datos relacionales. Los archivos grandes, los activos del juego o el contenido generado por los usuarios deben almacenarse en los servicios de almacenamiento de objetos.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Los juegos producen y consumen una gran variedad de tipos de datos que requieren soluciones de almacenamiento optimizadas en función de la frecuencia de acceso, la latencia y el coste. Los datos almacenados deben clasificarse mediante etiquetas para diferenciar los datos que se pueden eliminar o que deben almacenarse a largo plazo.

Los siguientes servicios funcionan bien para una variedad de casos de uso de los juegos:

[Amazon Aurora](#) (compatible con MySQL y PostgreSQL) ofrece alta disponibilidad, baja latencia y escalado automático, lo que lo convierte en una opción excelente para gestionar grandes cantidades de datos transaccionales, como la administración y la autenticación de las cuentas de los jugadores, las economías del juego, las tablas de clasificación y las clasificaciones de los jugadores, la persistencia del estado del juego, la gestión de eventos y campañas y el despliegue multirregional y de alta disponibilidad.

[Amazon DynamoDB](#) es una base de datos NoSQL totalmente gestionada conocida por su baja latencia, alto rendimiento y escalabilidad perfecta, lo que la hace ideal para gestionar los datos de los jugadores en tiempo real, la gestión de sesiones, el inventario, la economía del juego, el estado de las partidas multijugador en tiempo real, la creación de partidas, el registro de eventos y el escalado para una audiencia global.

[Amazon DocumentDB](#) (compatible con MongoDB) proporciona un servicio de base de datos orientado a documentos escalable y de baja latencia, perfecto para almacenar datos

semiestructurados y flexibles, como el sistema de inventario, los perfiles y la personalización de los jugadores, los mundos de los juegos y el contenido generado por procedimientos, las interacciones sociales y de los jugadores, el análisis y el seguimiento del comportamiento, y los metadatos y configuraciones del juego.

[Amazon ElastiCache](#) admite el almacenamiento en caché en memoria con Redis o Memcached, lo que ofrece un acceso rápido a los datos y tiempos de respuesta reducidos, lo cual es fundamental para los juegos multijugador en tiempo real, donde la velocidad y el rendimiento son esenciales para una experiencia de usuario fluida. ElastiCache se utiliza en juegos para tablas de clasificación en tiempo real, administración de sesiones, almacenamiento en caché de metadatos de juegos, chat y mensajería dentro del juego, creación de partidas, análisis y telemetría en tiempo real y escalado para eventos de alto tráfico.

[Amazon Simple Storage Service \(S3\)](#) se puede utilizar para almacenar objetos como activos de juegos, vídeos, imágenes, archivos de registro de texto y mucho más. S3 es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos, seguridad y rendimiento líderes del sector.

Si ofrece varias clases de almacenamiento que permiten el acceso frecuente y poco frecuente a los datos, y un almacenamiento de archivos rentable. En el caso de los datos a los que se accede con frecuencia durante el desarrollo, los estudios deberían almacenar los objetos en [S3 Standard](#) para obtener una latencia baja y un rendimiento de alto rendimiento. En el caso de los datos que suelen pasar de calientes a fríos o viceversa, los estudios deberían investigar [S3 Intelligent-Tiering](#). Intelligent-Tiering supervisa los patrones de acceso de sus datos y los mueve automáticamente al nivel de acceso más rentable.

[Para los estudios que necesitan un alto rendimiento y baja latencia y están de acuerdo con vivir en una sola zona de disponibilidad, usen S3 Express One Zone](#). Esto replica los datos en una única zona de disponibilidad y puede mejorar las velocidades de acceso a los datos en comparación con el estándar S3. Para las necesidades de archivado profundo de datos históricos, Amazon también ofrece [Amazon Glacier](#). Las clases de almacenamiento de Amazon Glacier están diseñadas específicamente para el archivado de datos, lo que le proporciona un alto rendimiento, flexibilidad de recuperación y almacenamiento de archivos de bajo costo en la nube.

[Amazon Elastic Block Store](#) se puede usar para almacenar binarios, archivos ejecutables y configuraciones de servidores de juegos que sus servidores de juegos o repositorios de activos necesitan para funcionar. Debe capturar y eliminar los volúmenes no utilizados que no estén adjuntos a una instancia. EC2 Esto le ahorra los gastos de almacenamiento incurridos y, al mismo tiempo, reduce el uso de hardware y servicios innecesarios.

Pasos para la implementación

- Clasifique los datos de los juegos por tipo, necesidades de retención y frecuencia de acceso, etiquetando los datos para distinguir entre los requisitos de almacenamiento a corto y largo plazo.
- Utilice Amazon Aurora para los datos transaccionales, DynamoDB para los datos de los jugadores en tiempo real, DocumentDB para los datos semiestructurados ElastiCache y para el almacenamiento en caché de baja latencia de la información de juego que es crucial para el tiempo.
- Almacene los activos del juego, los registros y el contenido generado por los usuarios en Amazon S3, seleccionando las clases de almacenamiento adecuadas (por ejemplo, Intelligent-Tiering, One Zone y Glacier) en función de los patrones de acceso y las necesidades de archivado, y utilice EBS para los binarios y las configuraciones de los servidores de juegos con una administración regular de instantáneas.

GAMESUS01-BP02 Usa políticas de ciclo de vida o caducidad de TTL para eliminar juegos innecesarios, datos de usuario, archivos de registro o activos obsoletos.

Puede utilizar etiquetas y tipos de datos para crear políticas de ciclo de vida o bien utilizar los TTL para mover los datos al almacenamiento de archivos o eliminarlos por completo del servicio. Esto puede incluir configuraciones temporales, contenido archivado caducado y registros históricos que ya no se necesitan. La mayoría de los servicios admiten el etiquetado.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

En el caso de los datos almacenados en S3, puede utilizar políticas de ciclo de vida para trasladarlos a niveles de almacenamiento de acceso y archivado poco frecuentes. En una configuración del ciclo de vida de S3, puede definir reglas para pasar objetos de una clase de almacenamiento a otra para ahorrar costos de almacenamiento. Cuando desconoce los patrones de acceso de los objetos o si los patrones de acceso cambian con el tiempo, es posible realizar la transición de los objetos a la clase de almacenamiento S3 Intelligent-Tiering para un ahorro automático de los costos.

Amazon S3 admite un modelo en cascada para las transiciones entre clases de almacenamiento, tal y como se muestra en el siguiente diagrama.

Puede agregar acciones de transición a una configuración de S3 Lifecycle para indicar a Amazon S3 que elimine objetos al final de su ciclo de vida. Cuando un objeto llega al final de su vida útil en función de la configuración de su ciclo de vida, Amazon S3 realiza una acción de caducidad en función del estado de control de versiones de S3 en el que se encuentra el bucket:

- **Depósito no versionado:** Amazon S3 coloca el objeto en cola para su eliminación y lo elimina de forma asíncrona, lo que elimina el objeto de forma permanente.
- **Depósito habilitado para el control de versiones:** si la versión actual del objeto no es un marcador de eliminación, Amazon S3 añade un marcador de eliminación con un identificador de versión único. Esto convierte la versión actual en no actual, y el marcador de eliminación en la versión actual.
- **Depósito suspendido por control de versiones:** Amazon S3 crea un marcador de eliminación con null como ID de versión. Este marcador de eliminación reemplaza la versión de un objeto por un identificador de versión nulo en la jerarquía de versiones, lo que elimina el objeto de forma efectiva.
- Cuando añade una configuración de ciclo de vida a un bucket, las reglas de configuración se aplican a los objetos existentes y a los objetos que añade posteriormente. Por ejemplo, si hoy añade una regla de configuración del ciclo de vida con una acción de caducidad que hace que los objetos con un prefijo específico caduquen 30 días después de su creación, Amazon S3 pondrá en cola para eliminar los objetos existentes que tengan más de 30 días de antigüedad y que tengan el prefijo especificado.

Tiempo de vida (TTL) para DynamoDB es un método rentable para eliminar elementos que ya no son relevantes. TTL permite definir una marca de tiempo de vencimiento por elemento que indica cuándo deja de necesitarse un elemento. DynamoDB elimina automáticamente los elementos vencidos a los pocos días de su fecha de vencimiento, sin afectar al rendimiento de escritura.

- Para usar TTL, primero debe activarlo en una tabla y, a continuación, definir un atributo específico para almacenar la marca de tiempo de vencimiento de TTL. La marca de tiempo debe almacenarse en el [formato de tiempo Epoch de Unix](#) con nivel de precisión de segundos. Cada vez que se crea o actualiza un elemento, puede calcular el tiempo de vencimiento y guardarlo en el atributo TTL.
- El sistema puede eliminar los elementos con atributos TTL válidos y vencidos, normalmente a los pocos días de su caducidad. Puede seguir actualizando los elementos vencidos que no se hayan eliminado todavía, lo que incluye cambiar o eliminar los atributos TTL. Al actualizar un elemento vencido, le recomendamos que utilice una expresión de condición para asegurarse de

que el elemento no se haya eliminado posteriormente. Utilice expresiones de filtro para eliminar los elementos vencidos de los resultados de [Scan](#) y [Query](#).

- Los elementos eliminados funcionan de forma similar a los que se eliminan con las operaciones de eliminación habituales. Una vez eliminados, los elementos pasan a DynamoDB Streams como eliminaciones de servicio en lugar de eliminaciones de usuarios, y se eliminan de los índices secundarios locales y de los índices secundarios globales al igual que otras operaciones de eliminación.

Con ElastiCache Redis, puede controlar la actualización de los datos almacenados en caché mediante el uso o la caducidad de las claves almacenadas en caché. TTLs Una vez transcurrido el tiempo establecido, la clave se elimina de la memoria caché y es necesario acceder al almacén de datos de origen además de acceder a los datos actualizados.

- Dos principios determinan qué patrones de almacenamiento en caché se TTLs deben aplicar y cuáles son los tipos de patrones de almacenamiento en caché que se deben implementar. En primer lugar, es importante que comprenda la tasa de cambio de los datos subyacentes. En segundo lugar, es importante que evalúe el riesgo de que los datos obsoletos se devuelvan a su aplicación en lugar de a la versión actualizada.
- En el caso de los datos dinámicos que cambian con frecuencia, es posible que desee aplicar menos datos TTLs que caduquen a un ritmo de cambio que coincida con el de la base de datos principal. Esto reduce el riesgo de que se devuelvan datos obsoletos y, al mismo tiempo, proporciona un búfer para reducir las solicitudes de la base de datos.
- También es importante tener en cuenta que, aunque solo guardes los datos en caché durante minutos o segundos, en lugar de hacerlo durante más tiempo, utilizarlos adecuadamente en las claves almacenadas en caché puede mejorar el rendimiento y, en general, mejorar la experiencia del jugador con el juego. TTLs

Pasos para la implementación

- Utilice las políticas de ciclo de vida de Amazon S3 para hacer la transición de los objetos a niveles de acceso o archivado poco frecuentes y configure acciones de caducidad para eliminar los objetos innecesarios en función de las reglas del ciclo de vida.
- Active el tiempo de vida (TTL) en las tablas de DynamoDB para eliminar automáticamente los elementos caducados sin consumir el rendimiento de escritura, definiendo la marca de tiempo de caducidad en Unix.

- Configura ElastiCache las claves adecuadas en función TTLs de las tasas de cambio de los datos y de la tolerancia al riesgo de datos obsoletos, lo que facilita la actualización de los datos en caché y mejora la experiencia del jugador.

Hardware y servicios

GAMESUS02: ¿Cómo gestionas el uso de los recursos informáticos en el backend de tus juegos?

Los estudios deberían desarrollar una estrategia de procesamiento que utilice una combinación de diferentes tipos de procesamiento, servicios gestionados y un plan de ahorro para optimizar su uso. También debes optimizar la forma en que los servidores de juegos y los servicios de backend se integran en las instancias de cómputo para reducir la cantidad de recursos innecesarios.

Prácticas recomendadas

- [GAMESUS02-BP01 Seleccione los servicios gestionados para las cargas de trabajo informáticas adecuadas](#)
- [GAMESUS02-BP02 Ajusta el tamaño de tu ordenador y despliega el rendimiento de la GPU solo donde sea necesario](#)

GAMESUS02-BP01 Seleccione los servicios gestionados para las cargas de trabajo informáticas adecuadas

Diseñe sus servicios de backend de juegos para utilizar servicios gestionados para cargas de trabajo de tráfico muy variables o impulsadas por eventos. Los servicios gestionados trasladan la gestión de la infraestructura a varios usuarios AWS y distribuyen el impacto ambiental entre varios usuarios debido a los múltiples planos de control de los inquilinos.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

AWS servicios como AWS Lambda AWS Fargate (Containers) y Amazon Gamelift (organización de servidores de juegos) pueden ejecutar código, contenedores u organizar sus servidores de juegos sin tener que administrar la infraestructura subyacente. Estos servicios se escalan automáticamente

en función de la demanda de los jugadores y solo se te cobrará por los recursos que consumas. Como la infraestructura subyacente se gestiona en tu nombre, puedes centrarte únicamente en los requisitos de tus juegos y servicios de backend.

Puede utilizar [AWS Lambda](#) puede ejecutar código sin aprovisionar ni administrar servidores. Lambda ejecuta el código en una infraestructura informática de alta disponibilidad y se encarga de la administración de los recursos informáticos, incluido el mantenimiento del servidor y del sistema operativo, el aprovisionamiento de capacidad, el escalado automático y el registro. Con Lambda, debe proporcionar el código en uno de los lenguajes de ejecución compatibles con Lambda. Lambda es útil para procesar eventos de juegos, autenticación de jugadores, procesamiento de compras dentro del juego y solicitudes de emparejamiento. Lambda escala automáticamente en función del número de eventos y puede gestionar picos de tráfico inesperados.

[AWS Fargate](#) es un motor de cómputo sin servidor para contenedores que funciona tanto con [Amazon Elastic Container Service](#) (ECS) como con [Amazon Elastic Kubernetes Service](#) (EKS). AWS Fargate permite centrarse fácilmente en la creación de aplicaciones, ya que reduce la necesidad de aprovisionar y administrar servidores, permite especificar y pagar los recursos por aplicación y mejora la seguridad mediante el aislamiento de las aplicaciones por diseño. Fargate es ideal para los servicios de backend que gestionan los perfiles de los jugadores, la gestión del estado y el emparejamiento.

[Amazon GameLift](#) es un servicio gestionado para implementar, operar y escalar servidores de juegos dedicados para juegos multijugador basados en sesiones. Puedes implementar tu primer servidor de juegos en la nube en cuestión de minutos, lo que te permitirá ahorrar hasta miles de horas de ingeniería en el desarrollo inicial de software y reducir los riesgos técnicos que suelen provocar que los desarrolladores eliminen las funciones multijugador de sus diseños.

Pasos para la implementación

- Úselo AWS Lambda para cargas de trabajo basadas en eventos, como el procesamiento de eventos de juegos, la autenticación de jugadores, las compras dentro del juego y las solicitudes de emparejamiento, aprovechando su escalado automático y su administración sin servidor.
- AWS Fargate Desplieguelo con ECS o EKS para servicios de backend, como los perfiles de los jugadores, la administración del estado y el emparejamiento, lo que elimina la administración de servidores y mejora el aislamiento de las aplicaciones.
- Usa Amazon GameLift para implementar y escalar servidores de juegos dedicados para juegos multijugador basados en sesiones, lo que reduce el tiempo de desarrollo y la complejidad operativa.

GAMESUS02-BP02 Ajusta el tamaño de tu ordenador y despliega el rendimiento de la GPU solo donde sea necesario

Diseñe los servidores y el backend de sus juegos para utilizar los recursos informáticos de forma eficiente. El aprovisionamiento excesivo de recursos informáticos puede generar costes innecesarios y minimizar la cantidad de recursos inactivos o infrautilizados. Las instancias de GPU deberían usarse para respaldar iniciativas de desarrollo específicas, como las recompilaciones de HLOD en Unreal, o si los servidores de juegos las requieren por diseño. Esto reduce considerablemente el impacto medioambiental y los costes de sus cargas de trabajo.

Nivel de riesgo expuesto si no se establece esta práctica recomendada: alto

Guía para la implementación

Debe optimizar los servidores de juegos y los servicios de backend para utilizar varios tipos de EC2 instancias y el menor número de instancias necesario. Esto aumenta el número de instancias disponibles para satisfacer tus necesidades durante el desarrollo o el lanzamiento de tus juegos. También debes hacer coincidir el tipo de instancia con la carga de trabajo específica que vayas a implementar. Las instancias optimizadas para cómputo admiten una amplia gama de casos de uso, incluidos servidores de juegos y servicios de backend, como el matchmaking. Las instancias optimizadas para memoria están diseñadas para ofrecer un rendimiento rápido para las cargas de trabajo que procesan grandes conjuntos de datos en la memoria. Utilice las instancias de GPU según sea necesario para los requisitos de alto rendimiento, pero no para las tareas informáticas generales. Si es posible, diseñe sus servicios o servidores de juegos para que se ejecuten en ARM con [instancias AWS Graviton](#). Graviton es el tipo de instancia con mejor rendimiento y eficiencia energética disponible en el mercado. AWS También ofrecen un rendimiento y unos costes mejorados en comparación con los tipos de instancias x86.

Úselo [AWS Compute Optimizer](#) para identificar las configuraciones de AWS recursos óptimas, como los tipos de instancias de Amazon Elastic Compute Cloud (EC2), las configuraciones de volúmenes de Amazon Elastic Block Store (EBS), los tamaños de las tareas de los servicios de Amazon Elastic Container Service (ECS), las licencias de software comercial AWS Fargate, los tamaños de las memorias de AWS Lambda funciones y las clases de instancias de bases de datos de Amazon Relational Database Service (RDS), utilizando el aprendizaje automático para analizar las métricas de utilización históricas. Compute Optimizer proporciona un conjunto de herramientas APIs y una experiencia de consola para reducir los costos y aumentar el rendimiento de las cargas de trabajo al recomendar los AWS recursos óptimos para sus cargas de trabajo. AWS

Pasos para la implementación

- Adapta los recursos de cómputo a cargas de trabajo específicas mediante instancias optimizadas para los servidores de juegos, instancias optimizadas para memoria para conjuntos de datos de gran tamaño e instancias de GPU solo para tareas como las reconstrucciones de HLOD o los servidores de juegos que dependen de la GPU.
- Optimice la utilización de la computación mediante la implementación de instancias AWS Graviton siempre que sea posible para lograr una mayor eficiencia energética, un mejor rendimiento y un ahorro de costes en comparación con las instancias x86.
- Úselo AWS Compute Optimizer para analizar el uso histórico y recomendar las configuraciones más eficientes para EC2 las cargas de trabajo de AWS ECS y Amazon RDS a fin de reducir los costos y mejorar el rendimiento. AWS Lambda

Recursos

Consulta los siguientes recursos para obtener más información sobre nuestras prácticas recomendadas relacionadas con la sostenibilidad.

- [ONU: La industria del juego destaca las amenazas al planeta](#)
- [Medio: La sostenibilidad ambiental en el desarrollo de juegos: jugar responsablemente para lograr un futuro más ecológico](#)

Servicios clave AWS

- [Amazon Aurora](#)
- [Amazon DynamoDB](#)
- [Amazon DocumentDB \(con compatibilidad con MongoDB\)](#)
- [Amazon ElastiCache](#)
- [Amazon S3](#)
- [Clases de almacenamiento de Amazon S3](#)
- [Clase de almacenamiento por niveles inteligente de Amazon S3](#)
- [Clase de almacenamiento Amazon S3 Express One Zone](#)
- [Amazon Elastic Block Store](#)
- [AWS Lambda](#)

- [Amazon Elastic Container Service](#)
- [Amazon Elastic Kubernetes Service](#)
- [Amazon GameLift](#)
- [AWS Compute Optimizer](#)

Conclusión

Los juegos están diseñados para ofrecer experiencias de entretenimiento a una audiencia global de jugadores y tienen características de uso que suelen ser impredecibles y variables. Desde el punto de vista de la industria de los videojuegos, se describen los tipos de escenarios habituales que suelen conformar una arquitectura de juegos y se ofrecen una serie de preguntas y prácticas recomendadas que hay que tener en cuenta a la hora de crear y utilizar juegos en la nube. Al aplicar este marco a la arquitectura de sus juegos, puede crear juegos confiables, seguros, eficientes y rentables en la nube.

Colaboradores

Las siguientes personas y organizaciones han colaborado en este documento:

- Adam Hatfield, arquitecto sénior de soluciones, Amazon Web Services
- Brady Webb, director técnico de cuentas de Amazon Web Services
- Bruce Ross, arquitecto sénior de soluciones, líder de Well-Architected Lens, Amazon Web Services
- Caleb Cecil, arquitecto de soluciones asociado, Amazon Web Services
- Carlos Pérez, éxito en la optimización de la nube SA, Amazon Web Services
- Chase Herrington, director técnico de cuentas de ESL en Amazon Web Services.
- Chris Blackwell, arquitecto sénior de soluciones, Amazon Web Services
- Corey Ouder Kirk, gerente técnico sénior de cuentas, Amazon Web Services
- Derek Villavicencio, Prototipado SA, Amazon Web Services
- Erik Ynigo Becerril, arquitecto sénior de soluciones, Amazon Web Services
- Grzegorz Ochmanski, arquitecto sénior de soluciones, Amazon Web Services
- Hadrian Baron, gerente técnico sénior de cuentas, Amazon Web Services
- Ian Armbruster, director sénior de soluciones al cliente de Amazon Web Services
- Jed O Bray, gerente técnico sénior de cuentas, Amazon Web Services
- Khurram Khokhar, gerente técnico sénior de cuentas, Amazon Web Services
- Kyle Somers, director sénior de arquitectura de soluciones, Amazon Web Services
- Madhuri Srinivasan, redactora técnica sénior, Well-Architected, Amazon Web Services
- Matthew Wygant, guía sénior de TPM, Well-Architected, Amazon Web Services
- Nataliya Godunok, éxito en la optimización de la nube SA, Amazon Web Services
- Nirav Doshi, arquitecto principal de soluciones, Amazon Web Services
- Olivia Liddell, arquitecta de soluciones, Amazon Web Services
- Randy James, director técnico principal de cuentas, Amazon Web Services
- Reou Ando, arquitecto de soluciones de juegos, Amazon Web Services
- Richard Raseley, gerente técnico sénior de cuentas, Amazon Web Services
- Sam Patzer, arquitecto sénior de soluciones, Amazon Web Services
- Scott Selinger, arquitecto sénior de soluciones, Amazon Web Services

- Sean Allen, arquitecto sénior de soluciones, Amazon Web Services
- Serge Poueme, arquitecto sénior de soluciones, Amazon Web Services
- Stewart Matzek, redactor técnico sénior, Well-Architected, Amazon Web Services
- Trenton Potgieter, arquitecto sénior de soluciones, Amazon Web AI/ML/Analytics Services

Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

Cambio	Descripción	Fecha
Nueva versión de lente	Toda la lente se actualizó con una nueva guía de mejores prácticas.	9 de diciembre de 2025
Publicación inicial	Documento técnico publicado por primera vez.	19 de noviembre de 2021

AWS Glosario

Para obtener la AWS terminología más reciente, consulte el [AWS glosario](#) de la Glosario de AWS Referencia.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.