AWS Livre blanc

Objectif de l'industrie du jeu

Objectif de l'industrie du jeu: AWS Livre blanc

Table of Contents

Lentille de l'industrie du jeu vidéo — AWS Well-Architected Framework

Date de publication : 9 décembre 2025 (Révisions du document)

Le AWS Well-Architected Framework aide les architectes du cloud à créer une infrastructure sécurisée, performante, résiliente et efficace pour leurs applications et leurs charges de travail. Basé sur six piliers (excellence opérationnelle, sécurité, fiabilité, efficacité des performances, optimisation des coûts et durabilité), Well-Architected propose une approche cohérente aux clients AWS et aux partenaires afin d'évaluer les architectures, de remédier aux risques et de mettre en œuvre des conceptions apportant une valeur commerciale.

Dans cette optique, nous nous concentrons sur la manière de concevoir, déployer et structurer les charges de travail de vos jeux dans le AWS Cloud. Nous définissons les composants, explorons les scénarios de charge de travail courants et définissons les principes de conception qui vous aideront à appliquer le Well-Architected Framework. Nous vous recommandons de commencer à concevoir votre architecture en tenant compte des meilleures pratiques et des questions du livre blanc AWS Well-Architected Framework. Ce document fournit des meilleures pratiques supplémentaires pour les clients de l'industrie du jeu vidéo.

Cet objectif définit les meilleures pratiques destinées à prendre en compte les caractéristiques uniques de la création et de l'exploitation de jeux dans le cloud, sur la base de notre expérience de travail avec des développeurs et des éditeurs de l'industrie du jeu vidéo dans le monde entier. Il fournit des conseils sur la manière de concevoir et d'exploiter votre environnement afin qu'il soit optimisé en termes de coûts et évolutif en fonction des fluctuations de la demande mondiale des acteurs. Cet objectif fournit également des conseils pour sécuriser votre infrastructure de jeu et optimiser les performances afin d'offrir une expérience de jeu positive.

Ce document est destiné aux personnes occupant des postes technologiques, tels que les directeurs technologiques (CTOs), les directeurs techniques des studios de jeux, les architectes, les développeurs et les membres de l'équipe opérationnelle. Après avoir lu ce document, vous comprendrez les AWS meilleures pratiques et stratégies à utiliser lors de la conception d'architectures pour les jeux.

Disponibilité de l'objectif

Les verres personnalisés étendent les conseils de bonnes pratiques fournis par AWS Well-Architected Tool. AWS WA Tool vous permet de créer vos propres lentilles personnalisées ou d'utiliser des lentilles créées par d'autres personnes qui ont été partagées avec vous.

Pour commencer à examiner votre charge de travail en matière de jeux, téléchargez et importez le Games Industry Lens dans le référentiel public AWS Well-Architected Tool d'objectifs AWS personnalisés Well-Architected. GitHub

Principes de conception

Le AWS Well-Architected Framework identifie les principes de conception généraux suivants afin de faciliter une bonne conception dans le cloud pour les charges de travail liées aux jeux :

- Comprenez le comportement des joueurs et leurs habitudes d'utilisation pour faire évoluer le jeu et protéger les joueurs : pour améliorer continuellement votre jeu et gérer efficacement l'expérience des joueurs, il est important de connaître la manière dont les joueurs interagissent avec le jeu lui-même et avec le reste des joueurs. Cela vous aide à comprendre comment améliorer le jeu, à gérer les coûts, à surveiller et à réagir aux activités d'utilisation non autorisées qui présentent des risques pour l'expérience des joueurs.
- Utilisez des technologies qui simplifient le fonctionnement du jeu et augmentent la vitesse de développement : privilégiez l'adoption de technologies susceptibles d'améliorer la vitesse et de réduire les frais opérationnels liés à la fourniture de nouvelles fonctionnalités et améliorations aux joueurs. Les jeux sont axés sur les succès et les joueurs ont de nombreux choix à prendre en compte. Il est donc essentiel d'agir rapidement et de s'adapter au changement pour le succès d'un jeu. Déterminez si vous êtes à l'aise avec l'utilisation de votre propre logiciel ou si vous préférez adopter un service géré comparable fourni AWS par AWS des partenaires ou les deux.
- Optimisez votre architecture pour améliorer les indicateurs qui reflètent l'expérience réelle des joueurs : au fur et à mesure que vous adapterez et ferez évoluer votre architecture, réfléchissez à l'impact de ces améliorations et changements sur l'expérience des joueurs. Les charges de travail des jeux doivent être capables de résister aux défaillances et de minimiser leur impact afin de bloquer les perturbations généralisées du gameplay. Les fonctionnalités de jeu et les systèmes qui ne sont pas étroitement dépendants les uns des autres doivent être découplés afin de réduire l'impact des échecs et d'isoler les problèmes ayant un impact sur les joueurs.
- Concevez l'infrastructure pour répondre aux pics de simultanéité des joueurs et évoluez dynamiquement en fonction des besoins : l'infrastructure doit être conçue pour s'adapter à la demande des joueurs. Les indicateurs, tels que la simultanéité des sessions des joueurs et le nombre de connexions, peuvent être utilisés pour effectuer une mise à l'échelle préventive avant que les systèmes ne soient surchargés. Les mesures réactives d'utilisation du système, telles que la consommation du processeur et de la mémoire, peuvent être utilisées pour évoluer en cas de surcharge des systèmes. En adaptant dynamiquement votre infrastructure, vous pouvez réduire les coûts d'exploitation de votre jeu.
- Implémentez des runbooks pour améliorer les opérations de jeu : les runbooks opérationnels doivent être utilisés pour gérer de manière cohérente les tâches récurrentes liées aux opérations

de jeu. Des runbooks devraient exister pour les flux de travail courants liés aux opérations de jeu, tels que la recherche et la réponse aux signalements des joueurs, la gestion des activités de pré-dimensionnement de l'infrastructure pour préparer les événements de grande envergure prévus tels que le lancement de nouvelles saisons et les sorties de contenu de jeu, et pour traiter les activités classiques de maintenance des jeux.

Scénarios

Dans cette section, nous abordons plusieurs scénarios courants dans une architecture de jeu. Chaque scénario inclut les caractéristiques communes qui sous-tendent la conception et un exemple de schéma d'architecture de référence.

Scénarios

- Hébergement de jeux pour un gameplay synchrone en temps réel
- Backends de jeu
- Architecture de backend de jeu basée sur un serveur
- Développement de jeux dans le cloud (CGD)
- Pipeline d'analytique de jeu

Hébergement de jeux pour un gameplay synchrone en temps réel

Le gameplay synchrone en temps réel permet à deux joueurs ou plus de participer et d'interagir simultanément dans un jeu où l'état du jeu est partagé entre les joueurs connectés afin de créer une expérience aussi proche que possible d'une expérience en temps réel. Parmi les exemples de jeux synchrones, citons les jeux de tir à la première personne, les jeux en ligne massivement multijoueurs (MMOG), les jeux de sport et d'action, ou les jeux en ligne où deux joueurs ou plus doivent être connectés pour partager leur expérience de jeu en temps quasi réel.

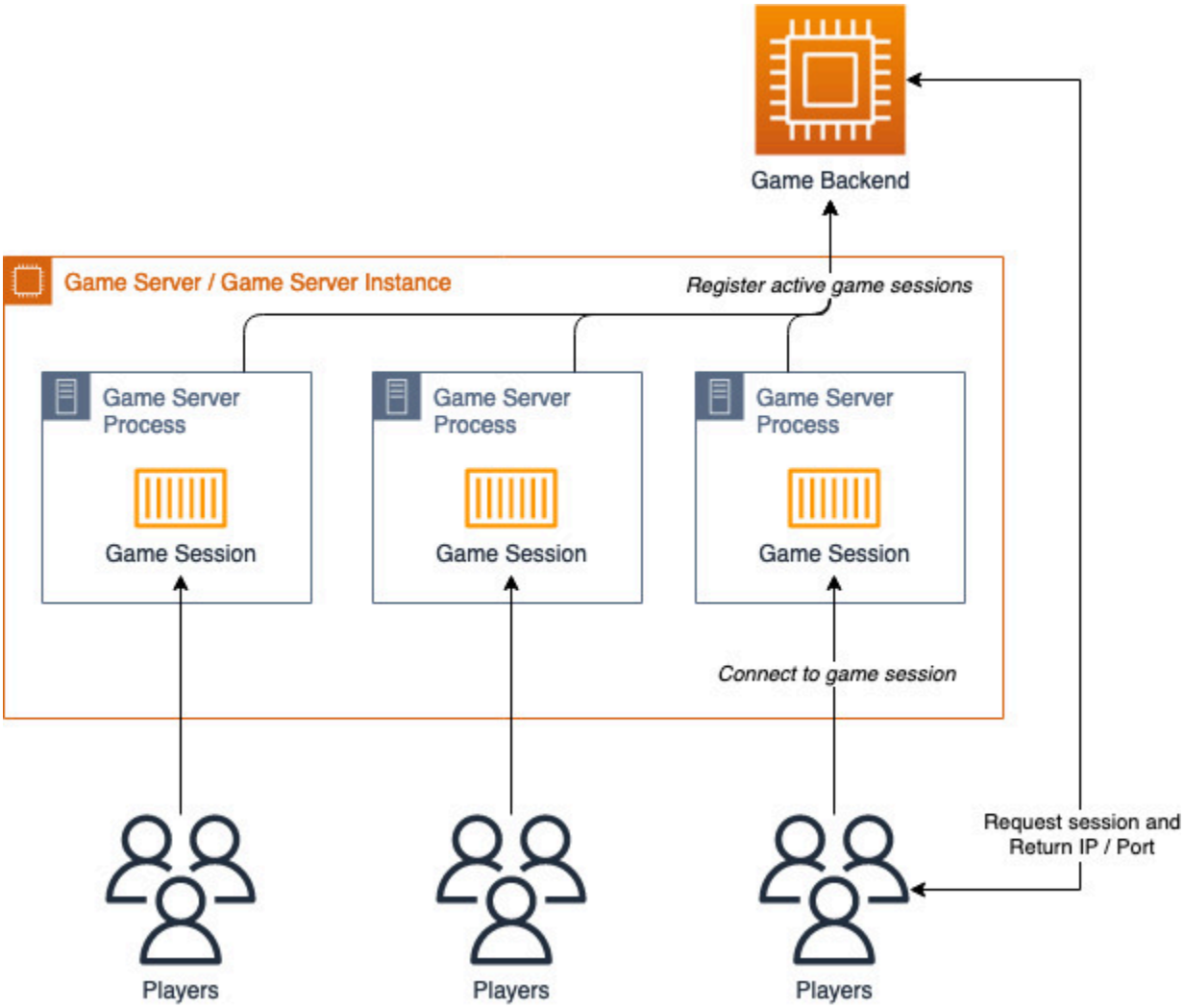
Les caractéristiques des architectures de jeu synchrones en temps réel incluent :

- Certains jeux peuvent être hébergés sous forme de sessions de jeu via des processus de serveur de jeu exécutés sur des serveurs dédiés. Certains jeux peuvent utiliser des architectures de type P2P qui utilisent des utilitaires de traversée de session plus légers pour les serveurs NAT (STUN) ou des utilitaires de traversée utilisant des relais autour de serveurs NAT (TURN). Quel que soit le type de serveur concerné, les serveurs de jeux sont hébergés dans plusieurs centres de données et dans Régions AWS le monde entier.
- Les clients du jeu peuvent rejoindre une session de jeu soit en demandant un match auprès d'un service de matchmaking centralisé hébergé dans le système principal du jeu, soit en sélectionnant un match dans une liste prédéfinie de serveurs de jeu disponibles. Le client du jeu dispose d'une adresse IP et d'un port auxquels se connecter.

- De nombreux jeux synchrones sont sensibles à la latence, comme les jeux de tir à la première personne et les jeux en ligne massivement multijoueurs. Ils incluront probablement des algorithmes tels que le retour en arrière et la dilatation du temps pour minimiser les effets de latence, mais ils peuvent également comporter une tolérance de latence prédéfinie qui est soigneusement mesurée et optimisée pour réduire le décalage qui peut parfois survenir pour les joueurs dans des situations de latence élevée. Ces informations de latence sont déterminées en instrumentant les clients du jeu pour qu'ils envoient un ping au serveur de jeu disponible Régions AWS afin de capturer des indicateurs tels que la latence, l'instabilité du réseau et d'autres indicateurs importants pour l'expérience de jeu. Ces statistiques sont envoyées à un service central de collecte de statistiques dans le système principal du jeu afin que les flux d'opérations en direct puissent surveiller l'état du jeu. Pendant le processus de matchmaking, les clients du jeu fournissent leurs données de latence actuelles comme l'un des paramètres de demande lorsqu'ils demandent un match, et le service de matchmaking peut utiliser ces données de latence comme l'une des variables lors de la sélection d'un serveur de jeu pour héberger le joueur.
- Généralement, le jeu se déroule sur une combinaison de protocoles (comme les serveurs de jeu utilisant une messagerie UDP plus rapide avec matchmaking, authentification et autre trafic client-serveur utilisant le protocole HTTPS).
- Les serveurs de jeux utilisent des algorithmes et une conception visant à minimiser le trafic client-serveur, comme le streaming de transformations, les deltas et la compression de données.
- Les serveurs de jeux sont fréquemment la cible d'activités malveillantes et doivent être protégés par une solution de protection DDo S telle que AWS Shield Advanced.

Processus du serveur de jeu

Le schéma suivant illustre l'architecture typique d'un serveur de jeu. Il décrit la relation logique entre une instance de serveur de jeu et les processus du serveur de jeu qui hébergent les sessions de jeu.



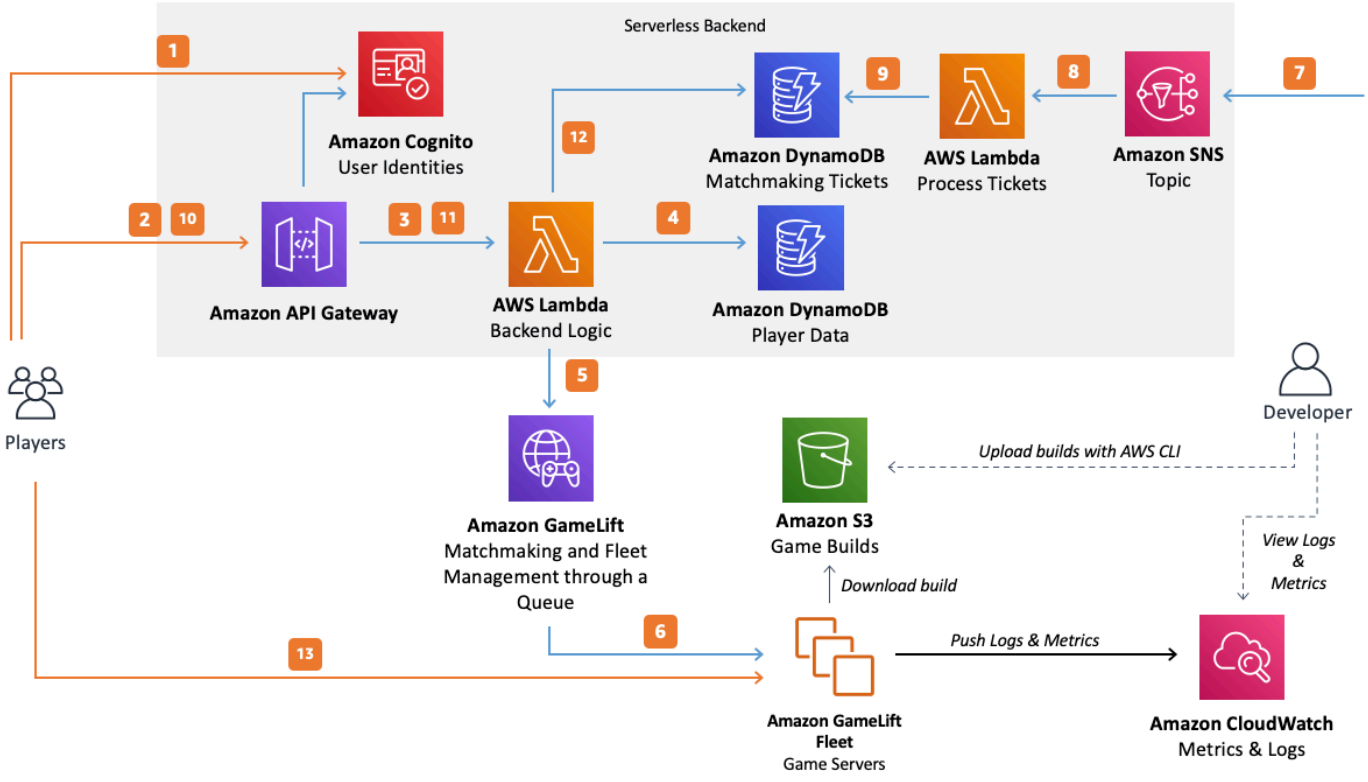
Architecture de serveur de jeu logique

- Une EC2 instance Amazon est utilisée comme serveur de jeu, également appelée instance de serveur de jeu. Le serveur de jeu héberge un ou plusieurs processus de serveur de jeu, et chacun exécute une copie de la version de votre serveur de jeu. Généralement, plusieurs processus de serveur de jeu sont exécutés sur une instance de serveur de jeu afin d'utiliser efficacement les ressources informatiques et de réduire les coûts. Lorsqu'une session de jeu est active et prête à héberger des sessions de joueurs, son statut est mis à jour avec le backend du jeu (généralement un service de matchmaking) afin qu'elle puisse commencer à être utilisée pour héberger des joueurs.
- Le backend du jeu peut fournir au joueur l'adresse IP et le port du serveur hébergeant une session de jeu afin qu'il puisse se connecter pour jouer.

Hébergement de serveurs de jeux basé sur des sessions avec backend sans serveur

Lorsque vous développez une architecture pour votre jeu, tenez compte des fonctionnalités et des capacités dont vous avez besoin, ainsi que du niveau de frais de gestion opérationnelle que vous êtes prêt à assumer. Pour trouver le meilleur équilibre entre facilité d'exploitation et flexibilité, vous pouvez créer votre jeu à l'aide des services gérés de fournisseurs de cloud. Les services gérés vous permettent de développer et de personnaliser vos propres fonctionnalités de jeu personnalisées, tout en réduisant la charge de déploiement et de gestion de l'infrastructure.

L'hébergement d'un jeu multijoueur basé sur des sessions nécessite de disposer d'une infrastructure de serveur pour héberger les processus du serveur de jeu ainsi que d'un backend évolutif pour le matchmaking et la gestion des sessions. L'architecture de référence suivante montre comment l'hébergement GameLift géré par Amazon et un backend sans serveur peuvent être utilisés pour gérer vos jeux basés sur des sessions.



Hébergement GameLift géré par Amazon pour les jeux basés sur des sessions

Le schéma décrit le processus d'introduction des joueurs dans des jeux exécutés sur un hébergement de jeux GameLift géré. Il comprend les étapes suivantes :

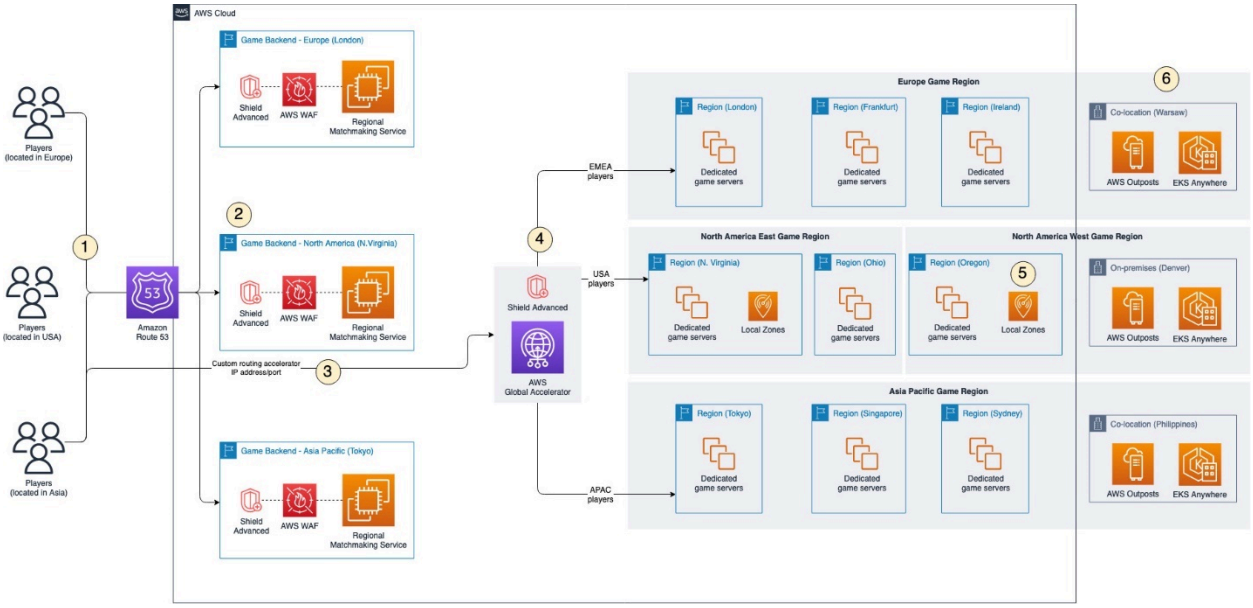
1. Le client du jeu demande une identité Amazon Cognito à un pool d'identités Amazon Cognito. Cela peut éventuellement être connecté à des fournisseurs d'identité externes.
2. Le client du jeu reçoit des informations d'accès temporaires et demande une session de jeu via Amazon API Gateway en signant la demande avec les informations d'identification Amazon Cognito.
3. API Gateway invoque une AWS Lambda fonction.
4. La fonction Lambda demande les données du joueur à partir d'une table Amazon DynamoDB. L'identité Amazon Cognito est utilisée pour demander en toute sécurité les données de joueur correctes, car l'identité authentifiée est fournie dans les données contextuelles de la demande.
5. En utilisant les données du joueur correctes pour obtenir des informations supplémentaires (comme le niveau de compétence du joueur), la fonction Lambda demande un match via le GameLift FlexMatch matchmaking. Vous pouvez définir une configuration de FlexMatch matchmaking avec des documents de configuration basés sur JSON. Le client du jeu peut générer des métriques de latence en envoyant un ping aux points de terminaison des serveurs dans différentes régions, et les données de latence peuvent être utilisées pour prendre en charge le matchmaking basé sur la latence.
6. Après avoir FlexMatch affronté un groupe approprié de joueurs avec une latence appropriée dans une région, celui-ci demande le placement d'une session de jeu via une GameLift file d'attente. La file d'attente contient des flottes avec un ou plusieurs emplacements régionaux enregistrés.
7. Lorsque la session est placée sur l'un des sites de la flotte, une notification d'événement est envoyée à une rubrique Amazon SNS.
8. Une fonction Lambda recevra l'événement Amazon SNS et le traitera.
9. Si le message Amazon SNS est un MatchmakingSucceeded événement, la fonction Lambda écrit le résultat dans DynamoDB avec le port et l'adresse IP du serveur. Une valeur time-to-live (TTL) est utilisée pour s'assurer que les tickets de matchmaking sont supprimés de DynamoDB lorsqu'ils ne sont plus nécessaires.
10. Le client du jeu envoie une demande signée à API Gateway pour vérifier l'état du ticket de matchmaking à un intervalle spécifique.
11. API Gateway invoque une fonction Lambda qui vérifie l'état du ticket de matchmaking.
12. La fonction Lambda vérifie DynamoDB pour déterminer si le ticket a réussi. En cas de succès, la fonction Lambda renvoie l'adresse IP, le port et l'identifiant de session du joueur au client. Si le ticket échoue, la fonction Lambda envoie une réponse indiquant que le match n'est pas prêt.

13Le client de jeu se connecte au serveur de jeu via TCP ou UDP en utilisant le port et l'adresse IP fournis par le backend. Il envoie l'identifiant de session du joueur au serveur de jeu, qui le valide à l'aide du SDK Amazon GameLift Server.

Vous pouvez également modifier l'architecture précédente pour utiliser API Gateway WebSockets avec Amazon GameLift. Dans cette approche, la communication entre le client du jeu et le service principal de votre jeu s'effectue à l'aide d'une implémentation WebSocket basée. Cette implémentation peut être utilisée pour que la fonction Lambda du backend du jeu envoie un message côté serveur au client du jeu via un modèle de sondage plutôt que d'implémenter WebSocket un modèle de sondage.

Architecture multirégionale et hybride pour les jeux à faible latence

Cette section décrit une architecture multirégionale et hybride pour les jeux à faible latence.



Réduction de la latence grâce à l'accélération du réseau et au déploiement de serveurs de jeux dans le monde

1. Les joueurs d'un jeu disponible dans le monde entier peuvent venir de n'importe où. Lorsqu'un joueur demande une session de jeu ou un match, son client de jeu envoie une demande au service de backend du jeu enregistré auprès d'Amazon Route 53. Le routage basé sur la latence de Route 53 peut être utilisé pour acheminer le joueur vers le backend de jeu disponible le plus proche.

2. Le backend du jeu est déployé dans plusieurs populations Régions AWS les plus proches des joueurs. Chaque backend de jeu comprend un service de matchmaking régional qui trouve une session de jeu dans toutes les régions du jeu. Bien que la demande de matchmaking d'un joueur soit traitée par un service de matchmaking régional proche de chez lui, le matchmaking service est capable de rediriger un joueur vers une session de jeu dans une autre région de jeu, si nécessaire. Cette action améliore la résilience et les performances. En outre, chaque service principal de jeu utilise AWS WAF et fournit un filtrage Web AWS Shield Advanced de couche 7 et un contrôle des robots, ainsi que des protections contre les attaques par déni de service (DDoS) de distribution. Vous disposez de nombreuses options pour créer votre service de backend de jeu, telles que le mode sans serveur, les conteneurs, les EC2 instances ou l'hébergement du service de backend de jeu dans vos propres centres de données.
3. Pour améliorer l'expérience des joueurs en réduisant la latence et l'instabilité du réseau, un accélérateur de routage personnalisé est déployé à l'aide de AWS Global Accelerator, qui optimise automatiquement le routage du trafic entre le client du jeu et le serveur de jeu via le réseau AWS mondial. Vous configurez l'accélérateur de routage personnalisé pour mapper les ports d'écoute de Global Accelerator au port d' EC2 instance de votre serveur de jeu. Le client du jeu se connecte à l'adresse IP et au port de Global Accelerator qui servent de proxy et achemine de manière déterministe les joueurs vers l'adresse IP et le port du serveur de jeu appropriés hébergeant la session de jeu.
4. Votre jeu inclut des régions de jeu logique conviviales qui représentent un ensemble de sites hébergeant des serveurs de jeu géographiquement proches les uns des autres, tels que l'Amérique du Nord ou l'Asie-Pacifique. Pour réduire la latence et augmenter la couverture géographique, une combinaison de différentes solutions d'hébergement de serveurs de jeux peut être utilisée pour améliorer l'expérience des joueurs. Priorisez l'utilisation de ces sites Régions AWS dans la mesure du possible, car ces sites sont dotés de toutes les fonctionnalités et ont la plus grande capacité d'accueil.
5. Utilisez les Zones AWS Locales pour héberger des serveurs de jeu dans des zones géographiques de joueurs mal desservis où vous ne disposez pas d'installations d'hébergement existantes ou où aucune installation d'hébergement n' Région AWS est disponible.
6. Déployez AWS Outposts dans vos centres de données sur site et vos fournisseurs de colocation existants pour créer un plan de contrôle et une expérience de gestion fluides sur chaque site de déploiement à l'aide de racks entièrement gérés et de serveurs montés en rack. AWS Outposts sont également bénéfiques si vous ne disposez pas d'une capacité de serveur existante dans votre environnement sur site. Toutefois, si vous souhaitez une implémentation hybride avec un logiciel exécuté sur votre propre infrastructure de serveur existante, vous devez utiliser Amazon EKS Anywhere, qui vous permet de créer et d'exécuter des clusters Kubernetes sur votre propre

infrastructure avec une connectivité à Amazon EKS. Cela fournit une vue de console cohérente de vos clusters Kubernetes sur site.

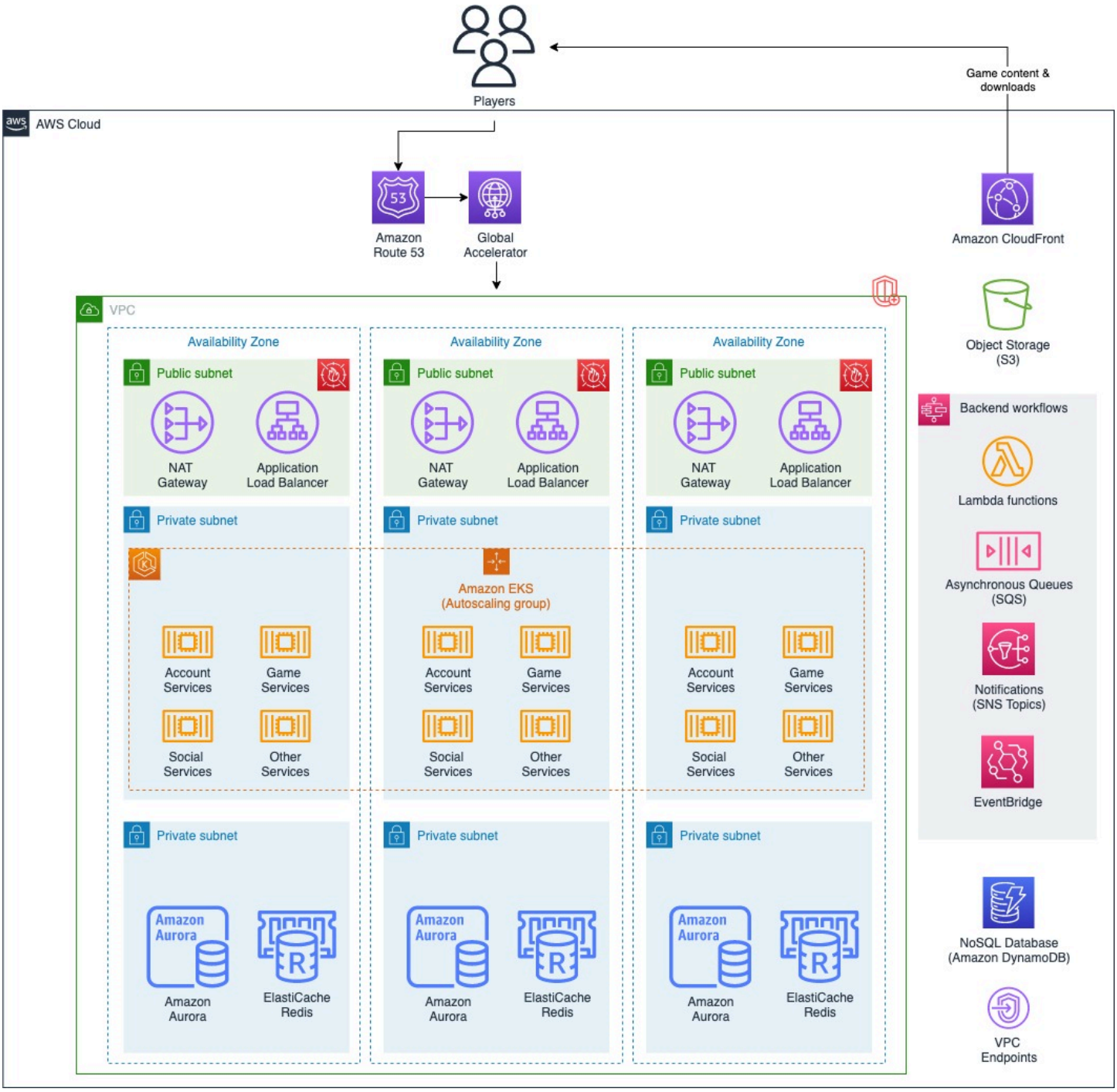
Backends de jeu

Les backends de jeu sont utilisés pour gérer le jeu et l'état des joueurs, ainsi que pour intégrer au jeu des fonctionnalités sociales et systémiques qui favorisent l'expérience de jeu. La gestion des profils des joueurs, le stockage des objets et de l'inventaire, les statistiques et les classements sont des exemples de services hébergés dans les backends de jeu.

Les backends de jeu sont généralement conçus en tant APIs que REST et sont accessibles aux clients via HTTPS. Cependant, d'autres approches sont également courantes, telles WebSockets que celles qui fournissent des canaux bidirectionnels pour les cas d'utilisation tels que les notifications aux clients pour le chat et la présence dans le jeu. Les backends de jeu peuvent être déployés à l'aide de différentes architectures de déploiement, notamment à l'aide d'instances, de conteneurs ou d'une architecture sans serveur.

Architecture de backend de jeu basée sur des conteneurs

Cette section décrit une architecture de backend de jeu basée sur des conteneurs.



Hébergement d'un backend de jeu à l'aide de conteneurs

- Les joueurs accèdent au jeu à l'aide d'un logiciel client de jeu, qui peut leur être distribué via un système de jeu, une vitrine numérique ou un téléchargement direct depuis un réseau de diffusion de contenu (CDN), tel qu'Amazon. CloudFront CDNsfournir une mise en cache à des emplacements périphériques afin d'améliorer les performances des utilisateurs qui téléchargent

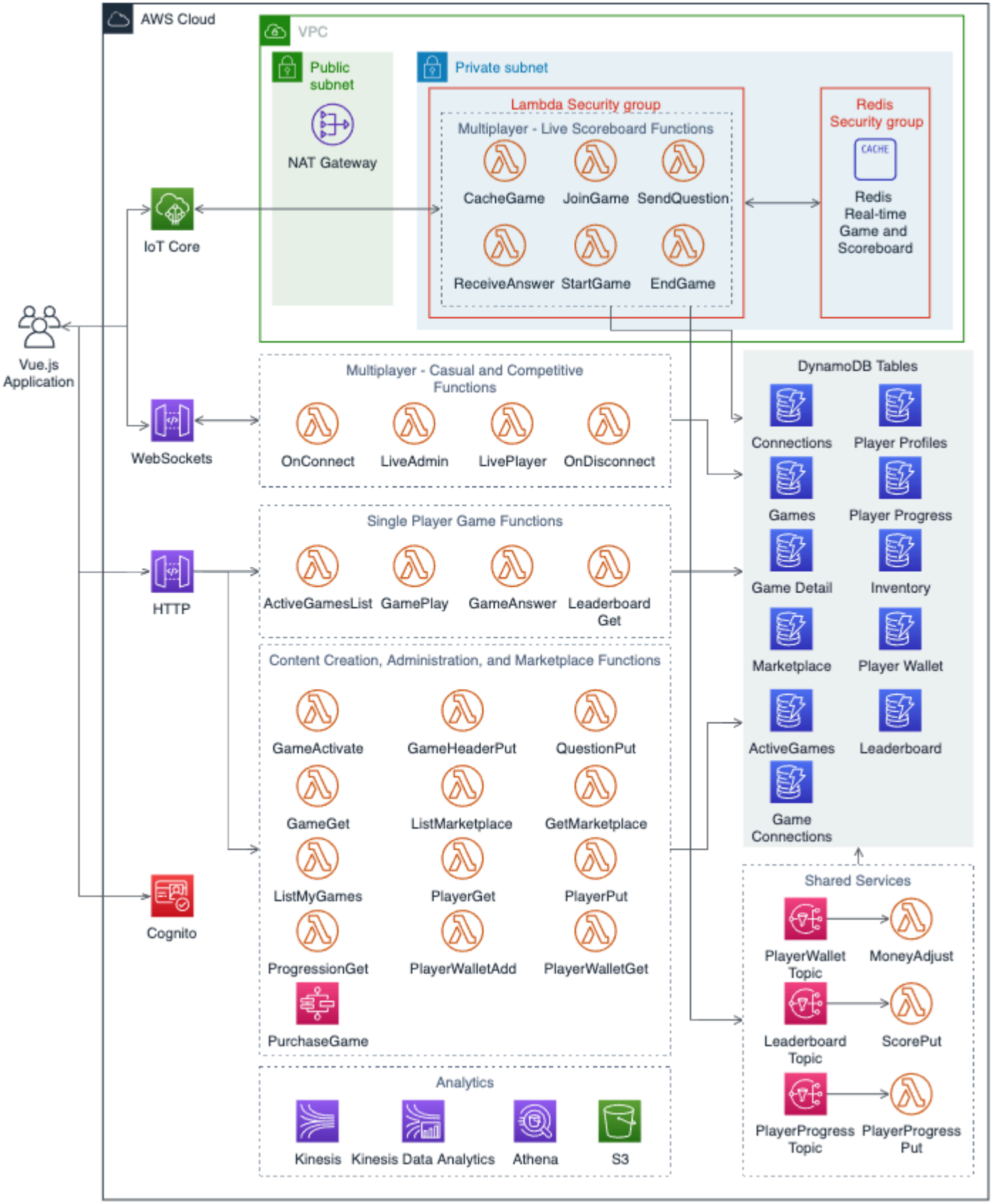
du contenu. Par exemple, CloudFront peut être utilisé pour distribuer le logiciel client du jeu à vos joueurs ainsi que les actifs du jeu et d'autres contenus.

- AWS Global Accelerator fournit une accélération du trafic et des commandes personnalisables pour acheminer le trafic des clients des jeux de joueurs vers vos équilibreurs de charge, ainsi que pour acheminer le trafic entre les régions à des fins multirégionales et de basculement. Les noms de domaine personnalisés pour votre backend de jeu REST APIs sont configurés dans Amazon Route 53 pour acheminer le trafic vers les points de terminaison Global Accelerator. Non illustré dans le schéma, cela AWS Shield Advanced peut fournir une atténuation DDo S supplémentaire pour votre accélérateur et votre backend de jeu.
- La passerelle NAT et l'Application Load Balancer sont déployés sur des sous-réseaux publics dans chacune des zones de disponibilité utilisées par le backend du jeu pour assurer une haute disponibilité dans la région.AWS WAF est déployé sur l'Application Load Balancer pour fournir un filtrage du trafic Web de couche 7.
- Le backend du jeu est hébergé sous la forme d'un ensemble de microservices individuels basés sur des conteneurs déployés dans un cluster Amazon EKS dans des sous-réseaux privés répartis dans des zones de disponibilité à des fins de résilience. Le dimensionnement automatique ajuste dynamiquement la capacité des services et des nœuds du cluster en fonction de l'utilisation des ressources, qui est généralement corrélée à la demande des joueurs. Alors que Cluster Autoscaler ajuste automatiquement le nombre de nœuds du cluster, le Horizontal Pod Autoscaler redimensionne automatiquement les pods déployés dans le cluster.
- Les données relatives aux jeux et aux joueurs sont stockées dans des bases de données et des caches principaux déployés dans des sous-réseaux privés répartis dans des zones de disponibilité, avec réplication entre les nœuds principaux et les nœuds de réplication. Amazon Aurora est un choix populaire pour les cas d'utilisation tels que les profils de joueurs, les droits et les achats intégrés au jeu, qui peuvent nécessiter des requêtes plus complexes et bénéficier des fonctionnalités de modélisation des données relationnelles de MySQL et PostgreSQL. Amazon ElastiCache est utile pour créer des classements et des pub/sub messages performants et pour mettre en cache les données fréquemment consultées afin de réduire la latence et la charge sur les bases de données. Amazon DynamoDB est un magasin de données NoSQL entièrement géré, idéal pour les modèles d'accès imprévisibles et capable d'évoluer jusqu'à un débit pratiquement illimité pour des cas d'utilisation tels que les données d'état des joueurs et des parties, les données de session, les stocks et les magasins d'articles, ou les cas d'utilisation où vous souhaitez disposer d'une base de données globale avec un minimum de surcharge.
- Les flux de traitement asynchrones doivent être utilisés pour effectuer des tâches pouvant être effectuées en arrière-plan, telles que la mise à jour des classements ou l'envoi de demandes

d'amis. Configurez le backend de votre jeu pour transférer ce type de travail dans les files d'attente Amazon SQS afin de l'adapter à la croissance de votre jeu, ou envisagez d'utiliser des rubriques Amazon SNS pour répartir le travail entre de nombreuses files d'attente d'applications grand public à des fins de traitement parallèle. Utilisez les fonctions AWS Lambda pour effectuer le traitement en fonction des événements afin de réduire les coûts de votre infrastructure informatique et les frais de gestion. Pour les flux de travail de longue durée ou nécessitant une coordination des tâches en plusieurs étapes, envisagez d'orchestrer l'ensemble du flux de travail à l'aide de.AWS Step Functions Amazon EventBridge peut être utilisé pour lancer des fonctions afin de répondre à des événements liés aux AWS services et aux applications personnalisées.

Architecture de backend de jeu basée sur un serveur

De nombreux développeurs de jeux ne souhaitent pas gérer l'infrastructure et préfèrent créer leurs jeux en utilisant des technologies qui leur permettent de se concentrer sur le logiciel. Une architecture sans serveur est recommandée dans ce scénario car elle vous permet de créer et de publier des fonctionnalités plus rapidement, tout en réduisant les frais opérationnels. Les architectures sans serveur sont conçues à l'aide de services cloud qui peuvent évoluer de manière dynamique en fonction de la demande sans avoir à configurer, gérer et dimensionner les serveurs. L'architecture de référence suivante illustre comment créer un jeu à l'aide d'une architecture sans serveur.


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

Architecture de référence du backend de jeu sans serveur

Cette architecture de référence illustre un jeu-questionnaire basé sur le Web qui propose des fonctionnalités solo et multijoueur.

- Authentification des joueurs : les joueurs s'authentifient à l'aide d'Amazon Cognito, qui fournit une authentification sécurisée avec un répertoire d'utilisateurs pour la gestion de l'identité des joueurs.
- La logique du jeu en tant que fonctions sans serveur : les fonctionnalités du jeu et la logique métier du backend fonctionnent comme des AWS Lambda fonctions lancées en réponse à des événements, ce qui permet de réduire les coûts car vous ne payez que lorsque la fonction est exécutée. Lambda vous donne la flexibilité d'écrire chaque fonctionnalité du jeu sous la forme d'un microservice distinct en utilisant le langage de programmation de votre choix. Par exemple, vous pouvez choisir de développer des fonctions Lambda .NET si vous avez déjà utilisé C# pour créer des jeux Unity, ou vous pouvez choisir de développer des fonctions Lambda Node.js si vous souhaitez programmer une interface et un backend pour un jeu Web à la fois. JavaScript
- Stockage de données NoSQL pour les données des jeux et des joueurs : utilisez DynamoDB pour stocker les données de vos joueurs et de vos jeux, car il est spécialement conçu pour stocker de grandes quantités de données provenant de microservices. Comme l'illustre cette architecture, il est recommandé d'utiliser des magasins de données distincts pour les besoins de stockage de données de chaque fonctionnalité du jeu, ce qui vous permet de surveiller et de gérer facilement les fonctionnalités de manière indépendante. Cela crée également des limites de séparation si la propriété des fonctionnalités ou des services change au sein de votre équipe. Dans cette architecture de référence, les tables DynamoDB sont utilisées pour stocker des données telles que l'état de connexion, les détails du jeu, la progression des joueurs et les informations du classement.
- Jeu solo : les fonctionnalités solo permettent aux joueurs d'effectuer des actions telles que sélectionner et jouer à un jeu et consulter le classement. Ces fonctionnalités sont mises en œuvre sous RESTful forme de services principaux hébergés par une API HTTP Amazon API Gateway qui invoque la fonction Lambda appropriée pour obtenir et définir des données dans des tables DynamoDB. Une fois le jeu terminé, le backend envoie également des notifications aux rubriques Amazon SNS qui lancent les fonctions Lambda de manière asynchrone afin de stocker la progression et les statistiques du joueur.
- Gameplay multijoueur : les fonctionnalités du jeu multijoueur nécessitent que les joueurs puissent interagir avec le jeu pour point-to-point communiquer, ainsi que pour diffuser et recevoir des mises à jour d'autres joueurs connectés. Une WebSockets implémentation est adaptée à point-to-point la communication dans un jeu léger, tel que des jeux-questionnaires. Les joueurs peuvent établir une WebSockets connexion à Amazon API Gateway WebSockets, qui gère la connexion et n'invoque

les fonctions Lambda que lorsqu'il y a des messages à envoyer ou à recevoir pour un joueur. Pour les cas d'utilisation où one-to-many la communication est requise entre les joueurs, AWS IoT Core fournit un support pour l'utilisation de la messagerie WebSockets via MQTT, qui permet aux clients de s'abonner à des sujets et d'agir sur les messages qu'ils reçoivent. Dans cette architecture, WebSockets les MQTT sont utilisés pour prendre en charge des cas d'utilisation tels que la diffusion de mises à jour en direct dans le jeu et la pose de questions aux joueurs connectés. Au lieu de cela AWS IoT, vous pouvez choisir Redis Pub/Sub pour la livraison des messages ou Redis Streams si vous avez besoin de conserver les messages.

- Utilisez les fonctions Lambda compatibles VPC pour accéder aux ressources de vos sous-réseaux privés : configurez les fonctions Lambda compatibles VPC pour accéder aux ressources des sous-réseaux privés de votre VPC, tels qu' ElastiCacheAmazon, qui est utilisé pour réduire les temps de requête pour les ensembles de données à faible latence tels que les classements en direct.

Pour plus d'informations, consultez les instructions relatives à l'hébergement de backend de jeux personnalisés sur AWS.

Développement de jeux dans le cloud (CGD)

Le développement de jeux dans le cloud (CGD) fait référence à l'infrastructure et aux outils nécessaires au cycle de vie du développement d'un jeu pour créer, tester et développer un jeu. Le développement de jeux est collaboratif entre les utilisateurs et les exigences en matière d'infrastructure changent fréquemment au cours des étapes de développement.

De nombreux développeurs de jeux font appel à des équipes de développement distribuées dans le monde entier et à distance, ce qui nécessite une technologie compatible avec ce type de développement. Les développeurs de jeux peuvent héberger tout ou partie de ces environnements AWS et utiliser la disponibilité mondiale de Régions AWS pour placer les ressources au plus près des utilisateurs et gérer leurs environnements de développement de manière plus rentable en adaptant les capacités de calcul et de stockage en fonction des besoins.

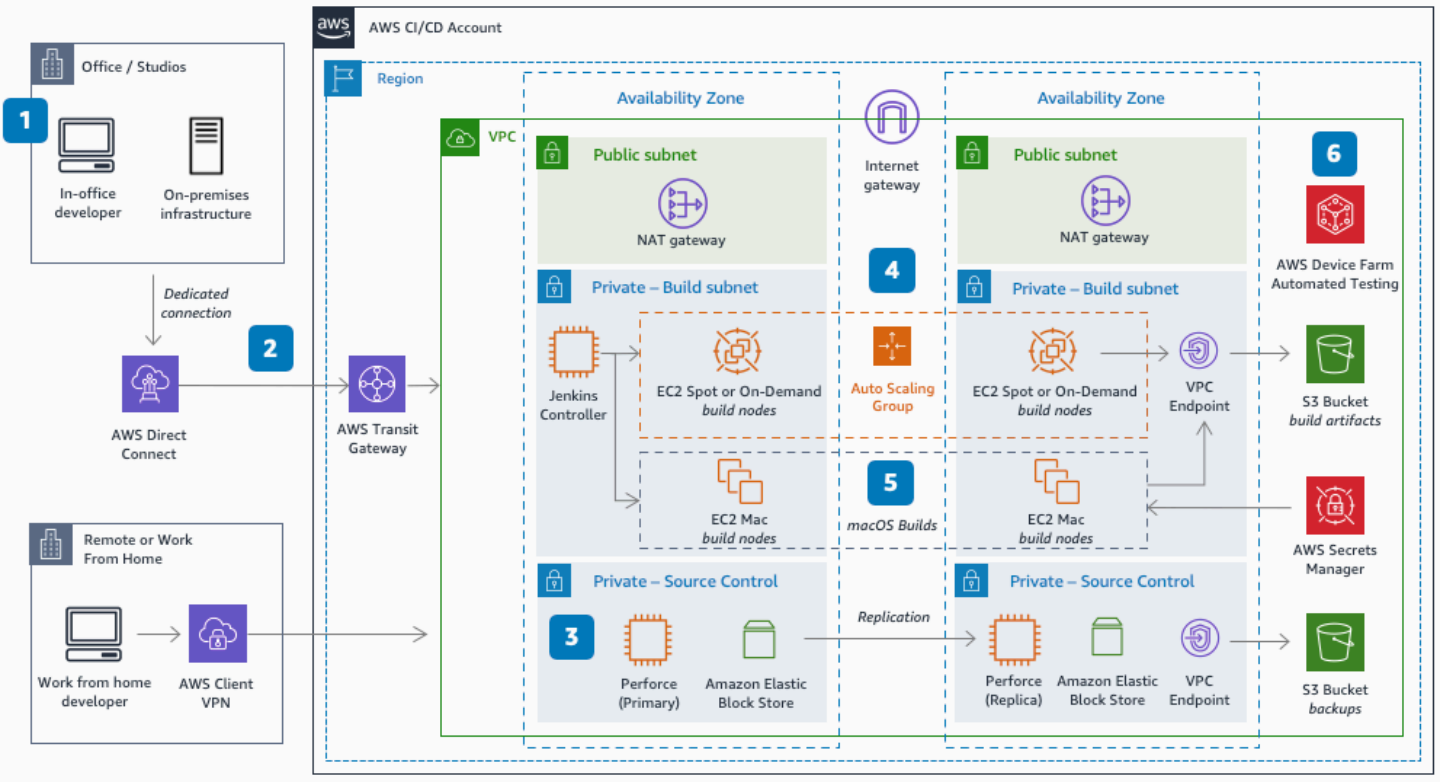
Les environnements peuvent varier en fonction des besoins des développeurs de jeux, mais ils incluent généralement des postes de travail pour les développeurs permettant aux artistes, aux concepteurs, aux ingénieurs, aux testeurs d'assurance qualité, aux sous-traitants et aux autres membres du personnel d'effectuer leur travail. Ces environnements incluent également généralement une ferme de compilation composée de référentiels de code source permettant aux utilisateurs d'enregistrer leurs modifications et d'une CI/CD infrastructure pour créer, empaqueter et tester les artefacts développés.

Ces architectures de production de jeux présentent les caractéristiques suivantes :

- Les utilisateurs doivent pouvoir accéder à une station de travail virtuelle via un navigateur Web ou un client de bureau local, tel qu'Amazon DCV, qui leur fournit une session de streaming à faible latence pour accéder aux mêmes logiciels et outils auxquels ils auraient accès s'ils travaillaient sur une machine dans un bureau ou un studio de développement. Ces postes de travail virtuels, généralement un serveur basé sur le cloud, devraient permettre à un utilisateur de collaborer et de travailler sur ses projets entièrement dans un environnement cloud via un réseau local ou étendu. Lorsque les utilisateurs n'utilisent pas activement les machines, leur travail doit être sauvegardé sur un stockage cloud durable, par exemple un référentiel de contrôle de source ou un système de fichiers tel qu'Amazon Elastic File System (EFS) et Amazon FSx, et leur machine doit être arrêtée pour réduire les coûts.
- Les référentiels de contrôle de source, tels que Perforce, doivent être conçus avec une haute disponibilité en utilisant la réplication entre des zones de disponibilité ou entre des environnements sur site avec des sauvegardes stockées dans un stockage cloud tel qu'Amazon S3. Par exemple, un serveur Perforce basé sur le cloud doit inclure un serveur de validation principal hébergé dans une zone de disponibilité avec réplication vers un serveur de secours hébergé dans une autre zone de disponibilité de la même région.
- Les ressources de la ferme de développement de jeux doivent être conçues avec une mise à l'échelle automatique afin que les ressources de calcul soient allouées selon les besoins, et les instances EC2 ponctuelles doivent être utilisées pour réduire les coûts liés à l'augmentation du nombre de serveurs requis pour les builds.

Développement de jeux sur le cloud : CI/CD

CI/CD l'infrastructure est importante lors du développement de jeux, quelle que soit la taille de l'équipe, afin d'améliorer les temps d'itération, de renforcer la fiabilité, d'assurer un déploiement efficace et de mieux contrôler le processus de développement et de publication afin d'offrir une expérience de jeu de haute qualité aux joueurs. Un CI/CD pipeline de développement de jeux est généralement composé de serveurs de contrôle de source et de stockage à haute disponibilité, de ressources informatiques pour exécuter vos builds et de logiciels pour effectuer des tests automatisés, ainsi que de la connectivité réseau appropriée depuis vos machines de développement. L'architecture de référence suivante montre comment décharger des builds de jeux depuis des environnements de développement de jeux distants ou sur site pour aider les développeurs AWS Cloud à migrer ou à créer de nouvelles fermes de versions.



Transférez les builds de jeux vers le cloud

1. AWS Direct Connect fournit une connexion privée dédiée à faible latence AWS aux développeurs internes. Les développeurs distants utilisent des technologies de confiance zéro ou des réseaux privés virtuels (VPN) tels que AWS le VPN Client. Accès vérifié par AWS
2. AWS Transit Gateway simplifie la gestion du réseau pour la connectivité entre VPCs et depuis les réseaux locaux.
3. Perforce gère le contrôle des sources et des versions (CI) soutenu par le stockage Amazon EBS pour des données persistantes accessibles rapidement. Perforce Helix Core est disponible dans le.AWS Marketplace
4. Les validations démarrent une compilation (CD) dans Jenkins lorsque les développeurs apportent des modifications à Perforce liées à une branche. Perforce lance POST une charge utile JSON vers Jenkins. Le contrôleur Jenkins appelle les commandes CLI headless du moteur pour exécuter et paralléliser le processus de génération sur des nœuds Docker éphémères (tels que les instances Amazon EC2 Spot ou les instances Amazon On-Demand). EC2 Les développeurs peuvent augmenter la disponibilité en utilisant deux contrôleurs Jenkins, un dans chaque zone de disponibilité, derrière un équilibreur de charge. Pour certains moteurs de jeu, les développeurs peuvent avoir besoin d'une infrastructure de licences supplémentaire configurée dans des sous-

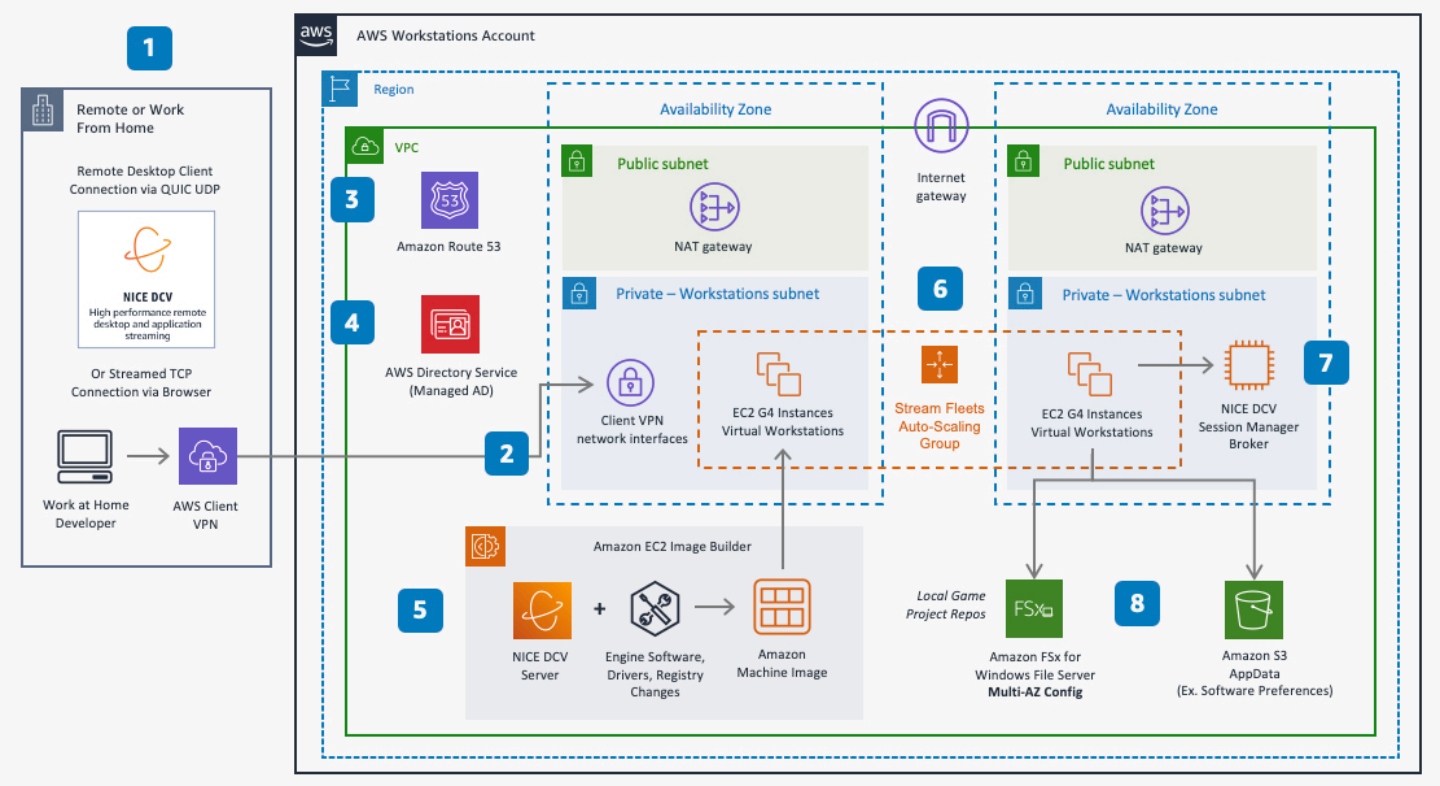
réseaux supplémentaires afin de vendre des licences pour le contexte de compilation chaque fois qu'une version simultanée est exécutée.

5. La partie Xcode des versions iOS est déchargée sur les instances Amazon EC2 Mac pour signer, créer et exporter le fichier .IPA, ce qui divise le processus et réduit les temps de création.AWS Secrets Manager contient des profils d'approvisionnement, des clés privées et des certificats.
6. Les artefacts de build sont transmis à Amazon S3, qui envoie des notifications de réussite ou d'échec. AWS Device Farm permet des tests automatisés pour les appareils mobiles.

Développement de jeux dans le cloud : Workstations

Le développement de jeux implique souvent des équipes distribuées travaillant à distance depuis différents sites, nécessitant un accès à une infrastructure partagée et la capacité de prendre en charge un développement collaboratif et géographiquement dispersé. Cette tendance vers un processus de développement de jeux plus distribué, y compris le recours à des work-for-hire studios et au travail à distance, nécessite la mise en œuvre de technologies et de flux de travail robustes pour faciliter la productivité, le partage d'expertise et des pratiques de développement agiles.

L'architecture de référence suivante montre comment l'utiliser AWS pour héberger des postes de travail de développement de jeux à distance à l'aide du protocole Amazon DCV.



Diffusez le développement de jeux où que vous soyez avec Amazon DCV

1. Amazon DCV est un protocole de streaming qui prend en charge le streaming 4K à 60 images par seconde. Les développeurs utilisant un navigateur se connectent via des connexions TCP, tandis que les clients de bureau peuvent utiliser QUIC UDP sur le port 8443 pour des performances accrues.
2. Les développeurs utilisent AWS le Client VPN pour établir une connexion sécurisée aux interfaces réseau des sous-réseaux des stations de travail avec traduction d'adresses réseau source (SNAT).
3. Amazon Route 53 fournit un DNS privé pour les ressources du VPC ainsi que le transfert DNS entrant et sortant.
4. Directory Service fournit Microsoft Active Directory géré pour permettre le stockage local de projets de jeux mappé à des utilisateurs individuels.
5. Les postes de travail sont créés à l'aide d'une Amazon Machine Image (AMI) créée avec Image Builder. Les images incluent le serveur Amazon DCV, les logiciels de développement, les modifications du registre et les pilotes tels que les pilotes de jeu ou les pilotes de périphériques NVIDIA. AWS Marketplace inclut les équipements couramment AMIs utilisés pour les postes de travail.

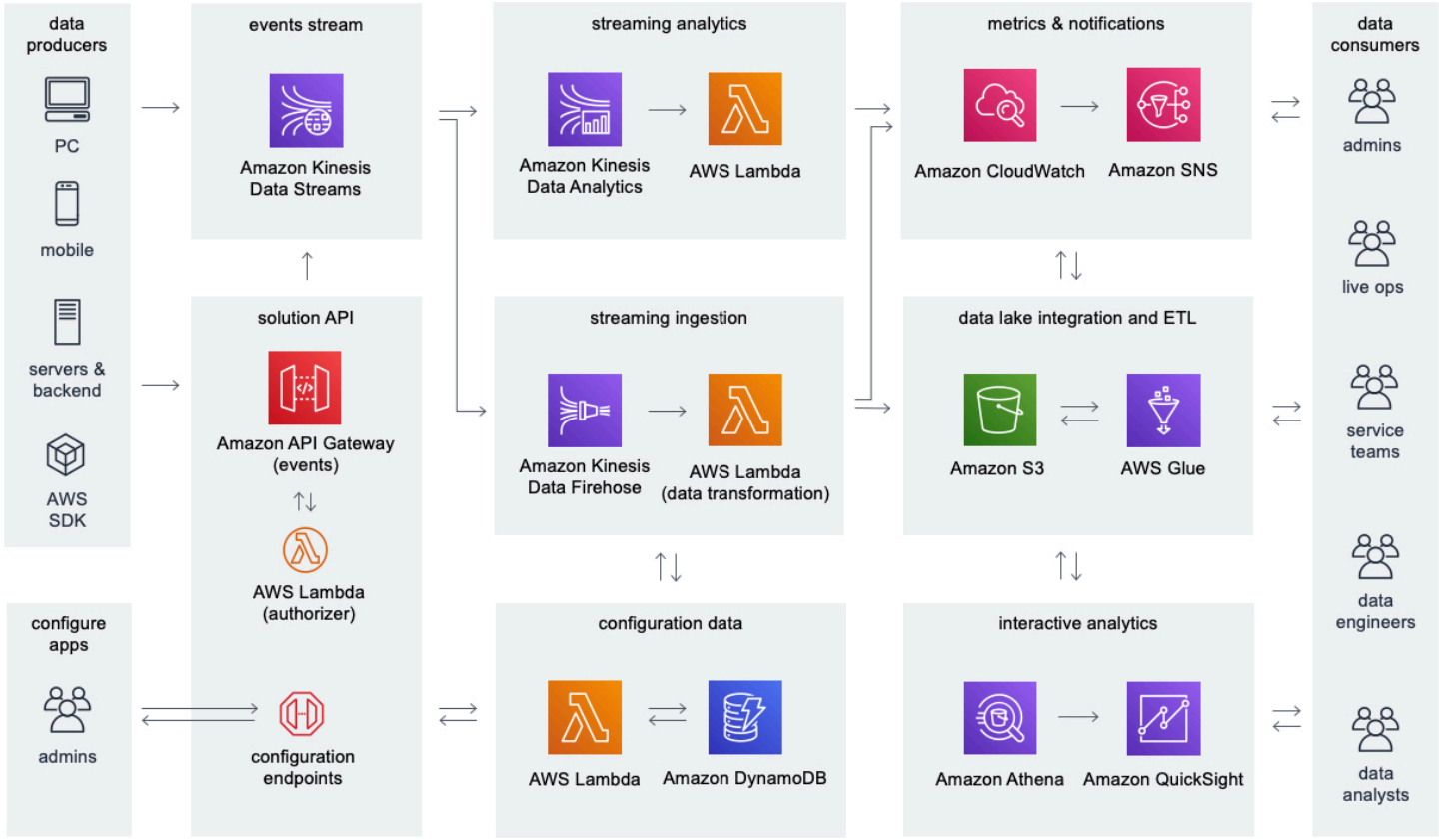
6. Les flottes de stations de travail utilisent des types d' EC2 instances Amazon graphiques qui fournissent des GPUs groupes EC2 Auto Scaling et sont dimensionnés à l'aide de ces groupes.
7. Le courtier Amazon DCV Session Manager permet de gérer les sessions Amazon DCV.
8. Le stockage local des fichiers des projets est hébergé sur Amazon FSx pour Windows File Server. Les développeurs s'engagent à mettre en place un pipeline CI/CD distinct en passant du stockage local au contrôle de source.

Pipeline d'analytique de jeu

Les développeurs de jeux cherchent de plus en plus des moyens de mieux comprendre le comportement des joueurs afin d'améliorer l'expérience de jeu afin de fidéliser et d'élargir leur base de joueurs. L'analyse des jeux représente l'infrastructure technique et les processus nécessaires pour comprendre et analyser les données générées par le jeu et les services associés. Cela nécessite généralement l'utilisation d'une architecture de pipeline d'analyse capable de prendre en charge ce end-to-end processus, telle que la mise en œuvre de la solution Game Analytics Pipeline.

Les architectures d'analyse de jeu présentent les caractéristiques suivantes :

- Les sources de données envoient des données dans un format courant tel que JSON et incluent généralement des serveurs de jeux et des services de backend, ainsi que des clients de jeux, notamment des PC, des appareils mobiles et des consoles de jeux.
- Un pipeline d'analyse de jeux automatise l'ensemble du flux de travail consistant à ingérer et à stocker les données brutes et à les traiter dans des formats de sortie utilisables afin qu'elles puissent être analysées de manière efficace et rentable par les consommateurs de données, tels que les utilisateurs finaux et les applications d'analyse.
- Les pipelines d'analyse des jeux permettent d'ingérer et de traiter de gros volumes de données en temps réel afin de les adapter à la croissance d'un jeu.
- Fournissez une assistance pour les cas d'utilisation des rapports en temps réel et par lots. Par exemple, les équipes opérationnelles utilisent généralement des tableaux de bord et des alertes en temps réel pour surveiller l'infrastructure du jeu et le comportement des joueurs afin de détecter les problèmes. Les équipes d'analystes de données s'appuient généralement sur des rapports par lots et selon les besoins pour comprendre les tendances au fil du temps.



Pipeline d'analyse de jeu sans serveur pour la télémétrie du jeu

Les données de jeu sont ingérées à partir des clients de jeu, des serveurs de jeux et d'autres applications. Les données de streaming sont ingérées dans Amazon S3 pour l'intégration des lacs de données et les analyses interactives. L'analyse du streaming traite les événements en temps réel et génère des métriques. Les consommateurs de données analysent les données métriques dans Amazon CloudWatch et les événements bruts dans Amazon S3.

- API de solution et données de configuration : utilisez Amazon API Gateway pour fournir une API REST permettant d'administrer votre pipeline d'analyse de jeux et de stocker les données de configuration dans Amazon DynamoDB à l'aide des fonctions Lambda. Vous pouvez créer un portail interne à partir de cette API ou une interface de ligne de commande personnalisée pour l'administration. Une API REST fournit également une authentification du serveur pour ingérer les données de jeu provenant de sources de données et transmettre les données de télémétrie à Amazon Kinesis Data Streams pour un traitement en temps réel et une ingestion dans le stockage.
- Flux d'événements : Amazon Kinesis Data Streams capture les données de streaming de votre jeu et permet le traitement des données en temps réel par Amazon Data Firehose et Amazon Managed Service pour Apache Flink.

- Analyse du streaming : le service géré pour Apache Flink analyse les données des événements de streaming provenant des Kinesis Data Streams et peut générer des métriques et des alertes personnalisées qui sont publiées à CloudWatch l'aide des fonctions Lambda.
- Mesures et notifications : utilisez Amazon CloudWatch pour surveiller les métriques, les journaux et les alarmes de votre solution. Utilisez Amazon SNS pour envoyer des notifications aux ingénieurs de garde et aux autres consommateurs de données.
- Ingestion du streaming : utilisez Firehose pour consommer vos données de streaming depuis Kinesis Data Streams et les transmettre à votre lac de données dans Amazon S3 pour un stockage, une transformation et une intégration à long terme avec d'autres données.
- Intégration du lac de données et ETL : à utiliser AWS Glue pour les flux de travail de traitement ETL et pour organiser vos métadonnées dans le AWS AWS Glue Data Catalog, ce qui constitue la base d'un lac de données à intégrer à des outils d'analyse flexibles.
- Analyses interactives : les utilisateurs finaux peuvent utiliser Amazon Athena pour effectuer des requêtes interactives ad hoc sur les ensembles de données stockés dans Amazon S3, et Quick Suite peut être utilisé pour créer des tableaux de bord.

Reportez-vous au Game Analytics Pipeline pour une implémentation de référence automatisée d'un pipeline d'analyse qui peut être déployé dans votre compte à l'aide de AWS CloudFormation.

Définitions

Le AWS Well-Architected Framework repose sur six piliers : excellence opérationnelle, sécurité, fiabilité, efficacité des performances, optimisation des coûts et durabilité. AWS fournit plusieurs composants de base qui vous permettent de concevoir des state-of-the-art architectures adaptées à vos charges de travail de jeu. Dans cette section, nous allons présenter un aperçu des principales définitions.

Pour les besoins de ce paper, une architecture de jeu englobe l'infrastructure technique principale requise pour créer et exploiter un jeu. Certains jeux peuvent ne pas avoir de fonctionnalités sociales, multijoueurs ou autres fonctionnalités en ligne et peuvent ne pas nécessiter l'utilisation de certains aspects de l'infrastructure technique du backend décrits dans ce paper. Pour une description détaillée des différents types de charges de travail fréquemment déployées pour prendre en charge une architecture de jeu, consultez la section Scénarios.

L' AWS Cloud infrastructure est construite autour de régions et de zones de disponibilité.

- Une région est un emplacement physique dans le monde où nous avons plusieurs zones de disponibilité.
- Les zones de disponibilité se composent d'un ou de plusieurs centres de données distincts, chacun doté d'une alimentation, d'un réseau et d'une connectivité redondants, hébergés dans des installations distinctes.

Selon les caractéristiques de votre jeu, vous souhaiterez peut-être déployer certains composants de votre architecture de jeu dans plusieurs régions pour des raisons telles que l'amélioration des performances pour les joueurs ou pour proposer des expériences personnalisées aux joueurs en fonction de leur situation géographique.

Il existe de nombreux types de jeux différents, et l'infrastructure technique requise pour prendre en charge un jeu varie en fonction du type de jeu développé. Par exemple, les types de jeux populaires peuvent inclure les jeux de tir à la première personne (FPS), les jeux de rôle (RPG), les jeux en ligne massivement multijoueurs (MMOG), les battle royales (BR), les jeux de sport, les jeux de puzzle, etc. Il existe également différents modes d'interaction qui influencent l'architecture du jeu, tels que le jeu au tour par tour et le jeu simultané, avec des caractéristiques de performance différentes.

Les jeux sont développés pour être joués sur un ou plusieurs systèmes de jeu, notamment les ordinateurs de bureau, le Web, les mobiles, les consoles, et les nouveaux modes d'interaction tels

que la réalité augmentée (AR), la réalité virtuelle (VR) et les solutions de streaming de jeux. Les jeux proposent généralement un gameplay multisystème, ce qui signifie que les joueurs peuvent enregistrer leur progression dans le jeu et reprendre le jeu sur d'autres systèmes, ainsi que lancer des sessions de jeu avec des joueurs sur d'autres systèmes.

La monétisation des jeux vidéo permet aux éditeurs de jeux de générer des revenus en utilisant différentes stratégies telles que la publicité, les achats de jeux numériques et de vente au détail, les achats intégrés de contenu téléchargeable (DLC) appelés microtransactions, et par le biais d'abonnements payants obligatoires pour jouer au jeu. Parmi les indicateurs de performance clés (KPIs) les plus courants dans l'industrie du jeu vidéo, citons :

- Daily active users (utilisateurs actifs quotidiens)
- Monthly active users (utilisateurs actifs mensuels)
- Utilisateurs simultanés (CCU)
- Durée de la session
- Coût par installation (CPI)
- Valeur de durée de vie du joueur (LTV)
- Revenu moyen par utilisateur (ARPU)

Système de jeu

Les jeux vidéo sont développés pour être joués sur un système de jeu qui fournit des commandes de saisie client, des graphismes, un logiciel client (connu sous le nom de client du jeu) et du matériel, et dans certains cas des fonctionnalités exclusives au système pour faciliter le jeu.

Les systèmes de jeu sont généralement répartis dans les catégories suivantes :

- Consoles : systèmes de divertissement spécialement conçus pour jouer à des jeux, y compris des modèles populaires tels que Sony, PlayStation Microsoft Xbox et Nintendo Switch. Les consoles permettent de jouer à des jeux en installant du contenu de jeu physique ou distribué numériquement sur le matériel de console fabriqué par le fournisseur du système de jeu. Dans cette définition, une console peut être portative ou fixe et destinée à être utilisée dans un scénario de divertissement à domicile.
- Ordinateur personnel (PC) : jeux joués à l'aide d'un logiciel informatique installé sur une machine cliente qui peut être personnalisé par le joueur. Pour cette raison, les jeux sur PC sont populaires auprès des joueurs en raison de la flexibilité et du contrôle qu'ils offrent.

- Web : jeux conçus pour être joués à l'aide d'un navigateur Web et qui offrent généralement l'avantage de permettre à un joueur d'accéder au jeu quel que soit son système d'exploitation.
- Mobile : jeux développés pour être joués sur un téléphone mobile, généralement un système d'exploitation pour smartphone. Les jeux mobiles sont généralement téléchargés depuis un magasin d'applications numérique et installés sur le téléphone.

Outre les systèmes mentionnés précédemment, il existe également des systèmes naissants qui sont encore relativement nouveaux et en pleine croissance et dont la part de marché est bien inférieure à celle des systèmes les plus prédominants. Les exemples de systèmes de jeu de cette catégorie incluent la réalité augmentée, la réalité virtuelle et le streaming de jeux, parfois appelés cloud gaming.

Le streaming de jeux consiste à rendre le gameplay dans le cloud et à le diffuser sur un client léger, généralement un navigateur. Le streaming de jeux permet à un joueur de jouer à un jeu entièrement hébergé à distance, généralement dans le cloud par un fournisseur de services de streaming de jeux. Dans le cadre du streaming de jeux, le joueur se connecte à un jeu basé sur le cloud via un navigateur ou un client léger fourni par le fournisseur de services de jeu cloud (système de jeu).

Serveur de jeu

Les serveurs de jeu représentent l'un des aspects les plus importants de l'infrastructure informatique de votre jeu. Les serveurs de jeu, parfois appelés serveurs de jeu dédiés, sont utilisés lors du développement d'un jeu multijoueur ou lorsque le traitement des événements de jeu faisant autorité par le serveur est requis. Le serveur de jeu est au centre de l'architecture du jeu, où s'exécute la logique de base, qui inclut la gestion du joueur et de l'état du jeu ainsi que la gestion des interactions entre les clients de jeu connectés et le serveur de jeu. Le serveur de jeu est généralement l'un des aspects les plus sensibles aux performances d'une architecture de jeu, car il est chargé de traiter les entrées du client de jeu d'un joueur et de les distribuer correctement aux autres joueurs connectés en temps réel. Un serveur de jeu peu performant a un impact sur les performances globales de l'expérience de jeu. Par conséquent, vous devez optimiser les performances du serveur de jeu et fournir une capacité suffisante, en particulier lors du lancement du jeu ou pendant les périodes de pointe.

Aux fins du présent document, un serveur de jeu ou une instance de serveur de jeu fait référence à l'ordinateur, tel qu'une machine virtuelle, qui héberge un ou plusieurs processus de serveur de jeu. Un processus de serveur de jeu représente une instance unique de la version de votre serveur de jeu hébergeant une session de jeu, qui est une instance de votre jeu en cours à laquelle les joueurs peuvent se connecter via une session de joueur. Pour cette raison, nous parlons souvent

de processus de serveur de jeu ou de session de jeu de manière interchangeable en raison de la relation individuelle implicite entre une session de jeu et le processus du serveur de jeu qui l'héberge. Il existe plusieurs options de calcul pour héberger des serveurs de jeu, qui donnent accès à une capacité évolutive basée sur le cloud grâce à un provisionnement élastique des ressources. AWS

Amazon EC2 fournit des serveurs virtuels basés sur le cloud, appelés instances, compatibles avec plusieurs versions de Linux et Windows. Vous pouvez créer des instances et les gérer directement comme un autre serveur ou une autre machine virtuelle. Généralement, plusieurs processus de serveur de jeu sont déployés sur une instance afin d'améliorer l'efficacité et de réduire les coûts. Amazon EC2 est un bon choix pour les serveurs de jeux si vous souhaitez avoir le plus de contrôle sur l'infrastructure informatique.

Amazon GameLift fournit une solution entièrement gérée pour l'hébergement de serveurs de jeux dédiés dans le cloud ainsi que des fonctionnalités supplémentaires telles que le matchmaking avec GameLift FlexMatch. GameLiftfournit une couche d'abstraction au-dessus d'Amazon EC2 pour simplifier la gestion des serveurs de jeux et est disponible dans la plupart des cas. Vous pouvez Régions AWS donc héberger des serveurs de jeux à proximité des joueurs afin de réduire la latence, d'atteindre une haute disponibilité et de réduire considérablement les coûts en utilisant des instances Spot. Bien qu'il GameLift puisse être intégré dans les backends de jeu existants, il est particulièrement utile pour les développeurs de jeux qui ne souhaitent pas développer leurs propres solutions de gestion de serveurs de jeu et de matchmaking et qui préfèrent une solution gérée AWS et évolutive au fur et à mesure que leur jeu grandit.

Amazon Elastic Container Service (Amazon ECS) est un service d'orchestration de conteneurs entièrement géré que vous pouvez utiliser pour exécuter des conteneurs basés sur Docker. Vous pouvez également utiliser Amazon Elastic Kubernetes Service (Amazon EKS) pour exécuter des conteneurs basés sur Docker créés à l'aide de Kubernetes. L'utilisation de technologies de conteneur telles que celles fournies par Amazon ECS et Amazon EKS peut vous aider à améliorer votre utilisation du calcul en regroupant efficacement de nombreux processus de serveur de jeu ou d'autres instances d'applications de jeu dans une EC2 instance.

L'utilisation de conteneurs peut également améliorer la productivité des développeurs en hébergeant des applications utilisant le même environnement d'exécution d'images Docker que celui utilisé par les développeurs sur leurs machines locales pendant le développement. Vous pouvez réduire davantage les frais d'exploitation en utilisant AWS Fargate une solution de calcul sans serveur pour l'exécution de conteneurs compatible avec Amazon EKS et Amazon ECS. Fargate convient parfaitement aux cas d'utilisation dans lesquels vous souhaitez exécuter des serveurs de jeux dans

des conteneurs sans être responsable de l'exploitation des instances sous-jacentes sur lesquelles les conteneurs s'exécutent.

Vous pouvez l'utiliser AWS Outposts pour exécuter AWS l'infrastructure et les services dans un centre de données ou une installation sur site, ce qui peut permettre aux jeux de fonctionner dans des environnements locaux et d' AWS utiliser la même infrastructure pour soutenir une stratégie d'adoption du cloud hybride. AWS Les Zones Locales sont des extensions Régions AWS qui permettent à vos serveurs de jeu et à d'autres charges de travail sensibles à la latence de fonctionner au plus près de vos joueurs ou de vos équipes de développement. En outre, pour réduire la latence du réseau mondial pour vos serveurs de jeu, vous pouvez utiliser AWS Global Accelerator pour améliorer les performances du trafic des joueurs vers vos serveurs de jeu.

AWS Lambda est un service de calcul sans serveur qui exécute du code sans provisionner ni gérer de serveurs, ce qui le rend utile pour les cas d'utilisation de serveurs de jeux asynchrones tels que les jeux au tour par tour ou ceux qui nécessitent des exigences de calcul légères, une base de code restreinte et où les fonctionnalités de jeu peuvent être conçues à l'aide d'une architecture de microservices sans état. Il est important de garder à l'esprit que les fonctions Lambda s'exécutent sur la base d'événements et par requête, plutôt que d'exécuter un processus de serveur de jeu de longue durée. Lambda fournit le plus d'abstraction d'exécution des options présentées dans ce paper, car les développeurs peuvent facilement choisir l'application sous-jacente pour héberger leur code.

Lorsque vous choisissez votre approche pour l'hébergement de serveurs de jeux, tenez compte de diverses exigences, notamment la surcharge opérationnelle, les bases de code existantes, les exigences de performance et l'évolutivité. EC2les instances et les conteneurs constituent de bonnes options pour les bases de code existantes, car ils nécessitent la moindre modification pour passer au cloud, et vous pouvez utiliser des EC2 instances pour dédier les ressources d'une instance de calcul, tandis que les conteneurs peuvent simplifier la gestion et le taux d'utilisation élevé. Les fonctions sans serveur offrent le plus haut niveau d'abstraction, que vous pouvez utiliser pour définir du code qui ne s'exécute qu'en réponse à des événements, ce qui peut réduire les coûts.

Client de jeu

Le client de jeu représente le logiciel et le matériel que le joueur utilise pour jouer à un jeu. Le client du jeu fournit le logiciel permettant de traduire les entrées du joueur en messages envoyés à un serveur pour traitement, et il est chargé de gérer les réponses entrantes du serveur et de rendre les résultats tels que les graphiques au joueur. Dans les jeux multijoueurs en réseau en temps réel, le client du jeu maintient généralement une connexion réseau permanente à un serveur de jeu pendant toute la durée d'une session de jeu afin de réduire la latence du réseau et de minimiser le temps de

traitement. Cependant, le client du jeu peut également interagir via REST avec un serveur de jeu ou des services principaux.

Messagerie

Il existe généralement trois catégories principales de messages dans les jeux :

- Messages d'engagement des joueurs destinés à un utilisateur ou à une cohorte d'utilisateurs spécifiques, tels que des invitations à des jeux ou des notifications push
- Messagerie de groupe entre joueurs, comme le chat en jeu
- ervice-to-serviceMessagerie S, telle que les messages JSON utilisés pour intégrer deux applications ou plus

Une stratégie courante pour envoyer et recevoir ce type de messages consiste à utiliser des modèles d'architecture de traitement asynchrone et éditeur-abonné. AWS fournit plusieurs services qui peuvent vous aider à implémenter la messagerie dans votre jeu.

- Amazon Simple Notification Service (SNS) : service géré permettant de distribuer des messages entre éditeurs et abonnés à l'aide d'un modèle d'architecture. pub/sub Les éditeurs envoient des messages via une API à Amazon SNS, qui les transmet de manière asynchrone aux applications abonnées et peut envoyer des notifications push directement aux clients mobiles ou aux ordinateurs de bureau grâce à la prise en charge de certains des services de notification push les plus utilisés. Amazon SNS peut être utilisé pour les notifications push destinées aux clients ainsi que pour les cas d'utilisation de service-to-service la messagerie.
- Amazon Simple Queue Service (SQS) : service de file d'attente entièrement géré qui facilite l'intégration des serveurs de jeu à votre jeu, quel que soit le langage de programmation utilisé dans chacun d'entre eux. De nombreuses tâches de jeu peuvent être découplées et gérées en arrière-plan, comme la mise à jour d'un classement ou des valeurs de temps de jeu dans une base de données. Cette approche est un moyen efficace de dissocier les différentes parties de votre jeu et de faire évoluer indépendamment les fonctionnalités destinées aux joueurs par rapport au traitement principal.
- Amazon Managed Streaming for Apache Kafka (MSK) : service entièrement géré qui simplifie la création de flux de données et d'applications destinées aux producteurs ou aux consommateurs à l'aide d'Apache Kafka, une solution open source populaire. Kafka est généralement utilisé pour ingérer et traiter des données de streaming en temps réel et peut être utilisé pour service-to-service la messagerie.

- Amazon ElastiCache (Redis OSS) : fournit un magasin de données en mémoire entièrement géré qui inclut la prise en charge de la pub/sub fonctionnalité populaire de Redis, couramment utilisée pour le développement d'applications de salon de discussion et de messagerie haute performance. service-to-service Redis prend également en charge les types de données riches tels que les listes et les ensembles afin que les développeurs puissent utiliser Redis pour des files d'attente de haute performance.
- Amazon Pinpoint : fournit des messages d'engagement des utilisateurs par e-mail, SMS, voix et notifications push. Par exemple, Amazon Pinpoint peut être utilisé pour envoyer des messages d'engagement des utilisateurs aux joueurs afin de les inviter à revenir au jeu et peut être utilisé pour des cas d'utilisation transactionnels tels que la prise en charge de jetons d'authentification multifactoriels, de confirmations de commande et d'e-mails de réinitialisation de mot de passe.

Opérations de jeu en direct (Live Ops)

Les opérations de jeu en direct (Live Ops) sont un style de gestion et d'opérations de jeu qui traite un jeu comme un service en direct et propose en permanence de nouvelles fonctionnalités, des mises à jour, des promotions, des événements en jeu et des améliorations au jeu lancé afin d'améliorer l'expérience de la communauté des joueurs.

Traditionnellement, les jeux étaient fournis sous forme de produits plutôt que de services, et de nouveaux contenus et fonctionnalités étaient fréquemment intégrés dans les versions ou les suites ultérieures plutôt que dans le produit lancé. Grâce à l'approche Live Ops de la gestion du jeu, une équipe chargée des opérations de jeu peut lancer un jeu et maintenir l'engagement de la communauté de joueurs par le biais d'expérimentations, de promotions, d'événements intégrés au jeu et d'innovations visant à divertir les joueurs.

Bien que cette approche présente l'avantage de débloquer de nouvelles stratégies d'engagement des joueurs et de générer des flux de revenus récurrents, elle nécessite une expertise opérationnelle accrue. Par exemple, pour mettre en œuvre une stratégie Live Ops réussie, un développeur peut avoir besoin d'intégrer des services cloud ou d'exploiter sa propre infrastructure technique principale. Ils ont également besoin d'un moyen efficace d'identifier et de résoudre les problèmes qui surviennent dans le jeu ou au sein de la communauté des joueurs et qui peuvent avoir un impact négatif sur l'expérience des joueurs.

Excellence opérationnelle

Le pilier de l'excellence opérationnelle met l'accent sur les meilleures pratiques en matière de déploiement et d'exploitation de jeux basés sur le cloud à grande échelle. Il est important de mettre l'accent sur l'excellence opérationnelle afin de maintenir une expérience positive pour les joueurs et de mettre en œuvre des mesures préventives afin de se préparer aux problèmes qui ont une incidence sur leur expérience et de s'en remettre.

Domaines d'intérêt

- Principes de conception
- Opérations en direct
- Structure du compte
- Déploiements de jeux
- Surveillance de la santé
- Test de charge
- Optimisation au fil du temps
- Ressources

Principes de conception

Outre les principes de conception énoncés dans le livre blanc Well-Architected Framework, les principes de conception suivants peuvent vous aider à atteindre l'excellence opérationnelle dans la création et l'exploitation de jeux :

- Définissez des objectifs mesurables et réalisables pour les équipes chargées des opérations de jeu et adaptez-les si nécessaire : en raison de la nature axée sur les succès des jeux, il est difficile de déterminer à l'avance combien de joueurs joueront à votre jeu lors de son lancement ou quelles seront les attentes des joueurs à l'égard de vos opérations de jeu en cours. Il est important de définir des objectifs ambitieux mais réalisables avec les parties prenantes et de concevoir une approche qui puisse être étendue si votre jeu dépasse les prévisions et réduite pendant que les équipes de développement optimisent l'expérience des joueurs. Préparez et testez de manière adéquate à l'avance pour répondre à ces exigences et alignez vos parties prenantes commerciales et techniques sur les objectifs cibles du fonctionnement du jeu. Une fois les objectifs définis, les équipes de jeu peuvent atteindre un équilibre approprié entre les coûts et les performances lors de

la planification, de la conception, du provisionnement, des tests, du déploiement et de l'exploitation de l'infrastructure principale du jeu.

- Utilisez des livrets opérationnels pour planifier les activités liées aux lancements de jeux et aux événements spéciaux : les équipes chargées des opérations de jeu doivent se coordonner avec les parties prenantes de l'entreprise pour modéliser des projections relatives au pic de concurrence entre joueurs prévu lors des événements et effectuer une planification proactive pour prédimensionner la capacité de l'infrastructure à l'avance. En raison de la nature fluctuante du trafic de joueurs pendant les événements, les activités de planification et de pré-dimensionnement préalables devraient renforcer vos systèmes de dimensionnement automatisés existants afin d'améliorer vos chances de succès lors d'un événement et de vérifier que vous disposez de suffisamment de ressources pour offrir une expérience positive aux joueurs. Mettez en œuvre des pratiques d'ingénierie des performances pour développer une base de référence de vos ressources et une compréhension basée sur les données de la capacité de votre système, ce qui aidera à orienter les activités de pré-dimensionnement et les configurations de dimensionnement automatisées. Développez des runbooks opérationnels pour assurer la cohérence du processus. Cette planification préalable et cette réactivité à la demande des joueurs sont particulièrement essentielles pour les jeux en direct, qui doivent maintenir des performances et une infrastructure fiables pour soutenir une base de joueurs actifs et engagés sur une longue période.
- Établissez un modèle opérationnel pour recevoir, étudier et répondre aux demandes d'assistance des joueurs : après le lancement, surveillez les signalements de plaintes et de problèmes liés au jeu. Mettez en œuvre des systèmes appropriés pour interagir avec les joueurs de manière sécurisée et efficace afin de résoudre de manière adéquate les problèmes des joueurs, tels que les forums communautaires, les réseaux sociaux, le courrier électronique, les systèmes de billetterie, les centres d'appels ou les solutions de chatbot automatisés. Cela est particulièrement crucial pour les jeux en direct, qui nécessitent une communication continue avec la base de joueurs, une réactivité aux commentaires des joueurs pour répondre à l'évolution des besoins et le maintien d'une communauté engagée pendant une durée de vie prolongée.

Opérations en direct

GAMEOPS01 : Comment définissez-vous la stratégie des opérations en direct (Live Ops) de votre jeu ?

Développez une stratégie d'opérations en direct (Live Ops) pour votre jeu en fonction d'objectifs définis et de mesures de performance, en consultation avec les parties prenantes de l'entreprise.

Bonnes pratiques

- GAMEOPS01-BP01 Utilisez les objectifs du jeu et les indicateurs de performance commerciale pour développer votre stratégie opérationnelle en direct

GAMEOPS01-BP01 Utilisez les objectifs du jeu et les indicateurs de performance commerciale pour développer votre stratégie opérationnelle en direct

Consultez les parties prenantes commerciales, telles que les producteurs de jeux et les partenaires d'édition, afin de déterminer les objectifs et les indicateurs de performance d'un jeu. Cela peut vous aider à élaborer des plans pour la gestion du jeu, notamment en définissant vos fenêtres de maintenance, les calendriers de mise à jour des logiciels et de l'infrastructure, ainsi que les objectifs de fiabilité et de restauration du système.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Ces statistiques peuvent également vous aider à déterminer à quel stade du cycle de vie de votre jeu vous devez intégrer un Steam opérationnel en direct (Live Ops) pour surveiller l'état de santé du jeu, recueillir des commentaires directs sur le jeu et mettre en place des processus de sortie rationalisés et automatisés. Par exemple, un nouveau jeu peut attendre qu'une certaine échelle soit atteinte, mesurée en fonction du nombre de joueurs actifs, des revenus ou d'autres indicateurs, avant de mettre en place une équipe dédiée aux opérations en direct. Un studio de développement de jeux bien établi possède peut-être déjà une expérience opérationnelle en direct, peut-être pour ses autres jeux. Il lui suffira donc d'intégrer le nouveau jeu.

Étapes d’implémentation

- Vous pouvez définir les objectifs en matière de simultanéité des joueurs (CCU) et d'utilisateurs actifs quotidiens et mensuels (DAU et MAU) que l'infrastructure du jeu doit être capable de prendre en charge efficacement, vos budgets d'infrastructure, vos objectifs financiers et d'autres objectifs de performance, tels que la fréquence de publication du contenu et des fonctionnalités pour accroître l'engagement des joueurs. Ces objectifs et indicateurs alimentent les décisions relatives

à la conception du jeu, à la gestion des versions, à l'observabilité et au support nécessaires à l'efficacité des opérations.

- Votre jeu peut avoir pour objectif de publier de nouvelles mises à jour de contenu au moins une fois par mois, sans interruption pendant la sortie. Ces informations vous aident à définir votre stratégie de déploiement des versions et à coordonner la planification de la maintenance requise qui peut nécessiter des interruptions de service à d'autres moments du mois et contribuent à votre SLA de disponibilité.

Structure du compte

GAMEOPS02 : Comment structurez-vous l'hébergement de vos Comptes AWS environnements de jeu ?

Mettez en œuvre une stratégie multi-comptes pour isoler les différents environnements de jeu et améliorer la sécurité, l'efficacité opérationnelle et l'évolutivité. Utilisez-le AWS Organizations pour gérer les hiérarchies de comptes, appliquer des garde-fous aux comptes et appliquer des politiques de balisage et de balisage sur les ressources déployées.

Bonnes pratiques

- GAMEOPS02-BP01 Adopter une stratégie multi-comptes pour isoler les différents jeux et applications dans leurs propres comptes
- GAMEOPS02-BP02 Organiser les ressources de l'infrastructure à l'aide du balisage des ressources

GAMEOPS02-BP01 Adopter une stratégie multi-comptes pour isoler les différents jeux et applications dans leurs propres comptes

Concevez une structure de compte qui orienterait le déploiement de l'infrastructure afin de répondre aux besoins opérationnels, de sécurité et d'isolation de chaque environnement. Il est essentiel d'isoler l'environnement en restreignant l'accès à celui-ci et en autorisant uniquement l'utilisation des AWS services requis, les environnements de production étant verrouillés, tandis que les environnements de développement et de test sont indulgents pour permettre l'expérimentation. Il est fortement recommandé d'isoler davantage les principaux sous-systèmes de chaque environnement, ainsi que

les services communs utilisés par plusieurs environnements pour être hébergés et gérés de manière autonome Comptes AWS .

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Adoptez une stratégie multi-comptes AWS en isolant les différents environnements (tels que le développement, les tests, la mise en scène, la production et les services partagés) au cas par cas Comptes AWS, afin de réduire la portée des incidents. AWS Organizations Envisagez de gérer de manière centralisée la hiérarchie de vos opérations Comptes AWS afin de simplifier davantage les opérations, ainsi que de définir et d'appliquer de manière sélective des politiques au niveau du compte et de l'unité organisationnelle (au niveau de l'unité organisationnelle). En concevant une UO et une Compte AWS structure adaptées à vos besoins de développement et de flux de production, vous pouvez optimiser vos coûts et améliorer l'évolutivité.

- Adoptez une stratégie multi-comptes : isolez les environnements pour réduire le rayon d'incidents et simplifier les opérations.
- Utilisation AWS Organizations : Gérez les comptes de manière hiérarchique, appliquez des politiques et activez une gouvernance centralisée.
- Planifiez l'évolutivité : concevez des structures de comptes précises et mettez en œuvre des mesures de réduction des coûts pour la croissance future.

Étapes d’implémentation

Un système de jeu déployé dans AWS doit utiliser plusieurs comptes organisés de manière logique pour assurer une isolation adéquate, ce qui réduit la portée des problèmes et simplifie les opérations à mesure que votre infrastructure de jeu évolue. Comptes AWS cette infrastructure de jeu hôte est généralement regroupée dans les environnements logiques suivants :

- Les environnements de développement de jeux sont utilisés par les développeurs pour développer les logiciels et les systèmes du jeu.
- Les environnements de test ou d'assurance qualité (QA) sont utilisés pour effectuer des tests d'intégration, des tests d'assurance qualité manuels et d'autres tests automatisés qui doivent être effectués.
- Les environnements de mise en scène ou de pré-production sont utilisés pour héberger le logiciel complet afin que des tests de charge et de fumée puissent être effectués avant le lancement en production.

- Les environnements live ou de production sont utilisés pour héberger le logiciel et l'infrastructure en direct et pour gérer le trafic de production provenant des joueurs.
- Les environnements de services ou d'outils partagés donnent accès à des systèmes, logiciels et outils communs utilisés par de nombreuses équipes différentes. Par exemple, un référentiel de contrôle de source central auto-hébergé et une ferme de création de jeux peuvent être hébergés dans un compte de services partagés.
- Les environnements de sécurité sont utilisés pour consolider les journaux centralisés et les technologies de sécurité utilisées par les équipes qui se concentrent sur la sécurité du cloud.

En ce qui concerne l'infrastructure de jeu AWS, il est recommandé de créer des comptes distincts pour chaque environnement de jeu (développement, test, mise en scène et production), ainsi que des comptes pour la sécurité, la journalisation et les services partagés centralisés.

Généralement, les petits studios de développement de jeux qui gèrent un nombre limité de ressources d'infrastructure, généralement quelques centaines de serveurs ou moins, peuvent en créer un Compte AWS pour chacun de ces environnements (par exemple, un compte de production, un compte de développement et un compte intermédiaire). Toutefois, au fur et à mesure que votre infrastructure de jeu ou la taille de votre équipe s'agrandit au fil du temps, ce modèle simplifié risque de ne pas bien évoluer.

Lorsque vous configurez ces environnements, tenez compte du fait que de nombreux AWS services partagent les ressources et les Quotas de Service au niveau de l'API pour l'ensemble d'un compte au sein d'une région donnée. Cela doit être pris en compte lors de la détermination de la manière d'organiser les comptes de manière logique. Comptes AWS n'entraînent des coûts que pour la consommation des services qui y sont déployés. Cela permet donc de réduire efficacement la contention des ressources et les quotas de service, en particulier à mesure que votre jeu grandit et que de plus en plus de développeurs ont besoin d'y accéder pour créer et gérer des ressources.

Sur la base de notre expérience de travail avec de grands studios de développement de jeux qui exploitent généralement des milliers de serveurs avec des centaines de développeurs accédant aux ressources, nous vous recommandons de concevoir une structure de compte plus fine dans laquelle les applications prenant en charge votre jeu disposent de leurs propres comptes de développement, de test, de mise en scène et de production. Comme il est difficile et fastidieux de repenser votre stratégie AWS multicompte une fois que vous avez lancé votre jeu en raison de la complexité de la planification et de la migration des systèmes live, tenez compte de vos futurs besoins d'évolutivité lorsque vous déterminez la bonne structure multicompte.

Vous pouvez l'utiliser AWS Organizationspour établir une hiérarchie Comptes AWS, regrouper et définir des unités organisationnelles (OUs) afin de leur appliquer des politiques communes au niveau de l'unité organisationnelle par le biais de politiques de contrôle des services ()SCPs. AWS Organizations gère et gouverne de manière centralisée votre environnement au fur et à mesure que vous développez et faites évoluer vos ressources. Vous pouvez créer de nouveaux comptes et allouer des ressources par programmation, regrouper des comptes pour organiser vos flux de travail, appliquer des politiques aux comptes ou aux groupes à des fins de gouvernance et simplifier la facturation en utilisant un mode de paiement unique pour vos comptes. En outre, Organizations est intégré à d'autres services afin que vous puissiez définir des configurations centralisées, des mécanismes de sécurité, des exigences d'audit et le partage des ressources entre les comptes de votre organisation.

AWS Control Towerfournit un moyen simple de configurer et de gérer un environnement multi-comptes sécurisé, appelé zone d'atterrissage. Control Tower crée votre zone de landing zone en utilisant AWS Organizations, en apportant une gestion et une gouvernance continues des comptes, ainsi que les meilleures pratiques AWS de mise en œuvre basées sur l'expérience de travail avec des milliers de clients lors de leur migration vers le cloud. AWS Config, AWS Trusted Advisor, et AWS Security Hub CSPMsont des services qui fournissent une vue agrégée ou centralisée de l'hygiène de votre compte.

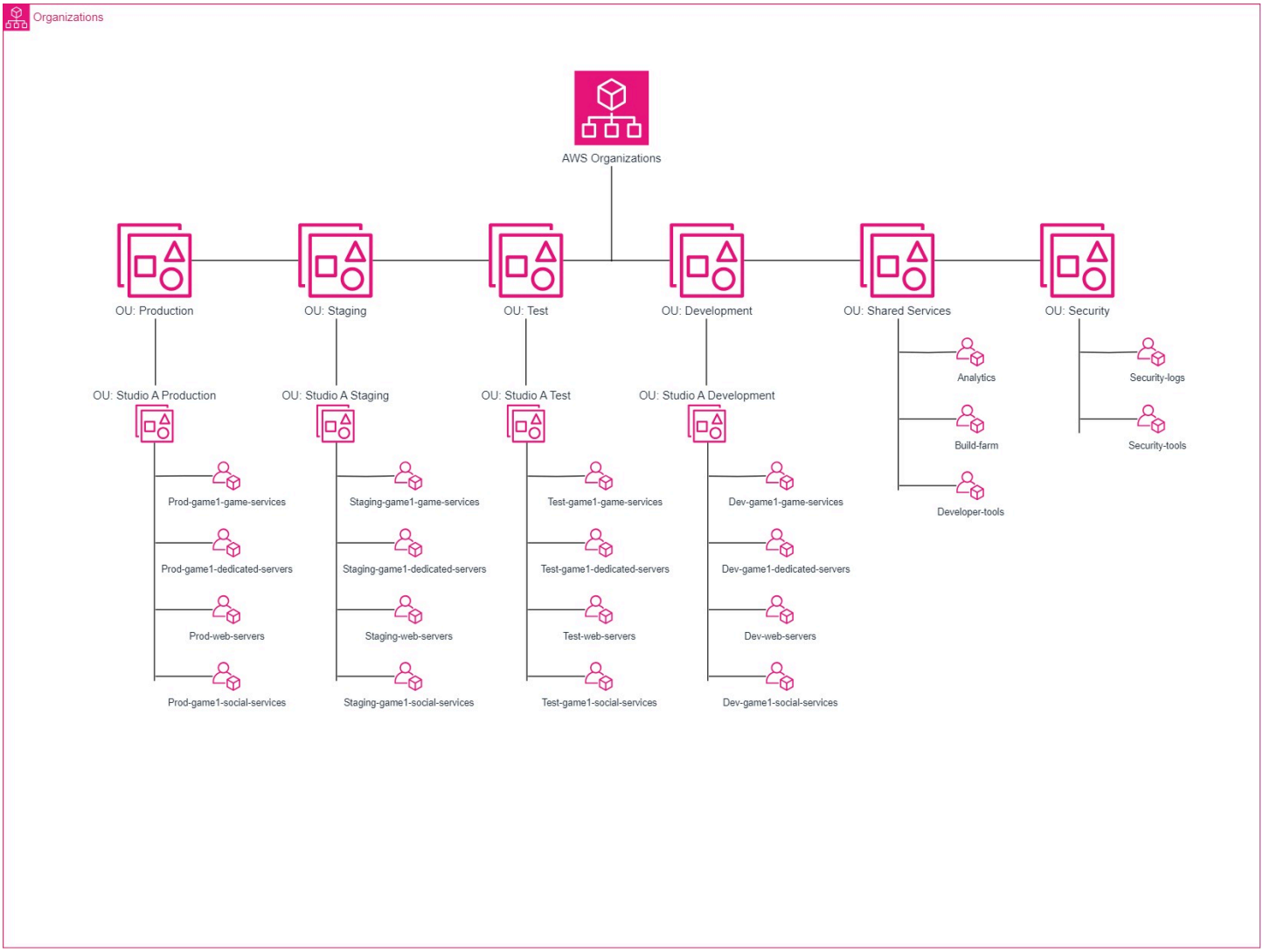
Cette isolation vous aide à configurer des autorisations et des garde-fous personnalisés ou individuels pour chaque environnement de jeu. Les comptes de production doivent disposer des garde-fous, des restrictions d'accès, des outils de surveillance et d'alerte et de sécurité nécessaires, tandis que les comptes hors production peuvent ne pas nécessiter le même niveau de garde-fous et d'autorisations. Les environnements hors production peuvent être automatisés pour arrêter les ressources en dehors des heures de bureau et réduire les coûts. La séparation des comptes à ce niveau de granularité facilite le suivi des coûts d'infrastructure pour chacun des environnements supportant un jeu.

Voici un exemple de structure multi-comptes pour une société de jeux vidéo utilisant AWS Organizations des unités organisationnelles (OUs) pour se regrouper logiquement Comptes AWS dans des environnements et des studios distincts. Dans cet exemple, OUs sont utilisés pour regrouper des comptes en fonction de leur environnement, puis en fonction du studio qui gère l'environnement. Cela montre comment vous pouvez créer une hiérarchie imbriquée pour permettre à des applications et à des jeux distincts d'être déployés dans leurs propres comptes au sein de leur environnement (illustré par OUs), ce qui peut être utile si vous développez et exploitez plusieurs jeux. Consultez la documentation et les livres blancs fournis dans la section des ressources

de ce pilier pour en savoir plus sur les stratégies supplémentaires que vous pouvez envisager pour organiser votre stratégie multi-comptes.

Sur la base de la discussion ci-dessus, l'exemple de schéma ci-dessous suppose un studio de jeu (organisation) doté d'un pipeline de développement composé de 4 étapes (développement, test, mise en scène et production). Pour un jeu donné (game1), chacun des environnements (UO) possède des services de jeu individuels Comptes AWS , des serveurs de jeu dédiés, des services sociaux et des serveurs Web. Les ressources qui s'exécutent dans chacun Compte AWS sont pertinentes pour les sous-systèmes respectifs. En général, chaque jeu utilisant ce type de pipeline de développement reproduirait cette structure ou une structure similaire pour le sien Comptes AWS.

Outre cet environnement centré sur le jeu OUs, il existe également l'unité d'organisation des services partagés et l'unité d'organisation de sécurité. Ils OUs devraient être applicables à l'ensemble de l'organisation, et non à chaque jeu individuel. De cette façon, les jeux consommeraient les services partagés pour les outils de développement, les données et les analyses, comme dans cet exemple. Envoyez ensuite les journaux de l'application et du système à la Compte AWS configuration des journaux dans l'unité d'organisation de sécurité.



Exemple de structure de compte pour les environnements de jeu

GAMEOPS02-BP02 Organiser les ressources de l'infrastructure à l'aide du balisage des ressources

Pour gérer et suivre efficacement les ressources de votre infrastructure AWS, utilisez un balisage et un regroupement appropriés des ressources pour identifier le propriétaire, le projet, l'application, le centre de coûts et les autres données de chaque ressource. Les ressources balisées peuvent être regroupées à l'aide de groupes de ressources, ce qui facilite le soutien opérationnel.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Définissez les politiques de balisage. Les stratégies classiques incluent des balises de ressources pour identifier le propriétaire de la ressource, telles que le nom de l'équipe ou le nom individuel, le nom du jeu, de l'application ou du projet, le nom du studio, l'environnement (comme le développement ou la production) et le rôle de la ressource (tel que le serveur de base de données, le serveur Web, le serveur de jeu dédié, le serveur d'applications ou le serveur de cache). Vous pouvez ajouter d'autres balises pour répondre aux besoins commerciaux et informatiques. AWS Configpeut également appliquer une politique de balisage au moment de la création et de la mise à jour des ressources. Les balises et les groupes de ressources sont disponibles à partir des opérations AWS CLI, the et API. AWS Management Console

Étapes d’implémentation

- Étiquetez les ressources pour identifier leur propriétaire, leur projet, leur application, leur centre de coûts et les autres données pertinentes.
- Mettez en œuvre des politiques de balisage, notamment des balises pour le propriétaire, le projet, le studio, l'environnement et le rôle de la ressource.
- AWS Config À utiliser pour appliquer les politiques de balisage et gérer les balises par le biais AWS Management Console de la CLI et de l'API.

Déploiements de jeux

GAMEOPS03 : Comment gérez-vous les déploiements de jeux ?

Gérez les déploiements de jeux en validant minutieusement les composants réutilisés, en effectuant régulièrement des études de performance et en mettant en œuvre des tests de charge réguliers tout au long du cycle de développement.

Bonnes pratiques

- GAMEOPS03-BP01 Validez et testez vos principaux systèmes et infrastructures de jeu existants avant de les réutiliser dans votre jeu
- GAMEOPS03-BP02 Procéder à l'ingénierie des performances avant chaque sortie (ou du moins pour les versions majeures)

- GAMEOPS03-BP03 Testez la charge tôt et souvent
- GAMEOPS03-BP04 Adopter une stratégie de déploiement qui minimise l'impact sur les joueurs
- GAMEOPS03-BP05 Infrastructure prédimensionnée requise pour répondre aux exigences de pointe

GAMEOPS03-BP01 Validez et testez vos principaux systèmes et infrastructures de jeu existants avant de les réutiliser dans votre jeu

Organisations ont tendance à réutiliser les composants existants et le code source des jeux précédents pour économiser du temps et des coûts de développement. Il est possible que ces composants et codes existants ne soient pas soumis à un examen approfondi ou à des tests d'intégration détaillés et qu'ils se basent plutôt sur leurs performances passées.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Bien que la réutilisation contribue à améliorer la productivité, elle peut également présenter le risque de réintroduire des problèmes de performance et de stabilité antérieurs dans un nouveau projet. Par conséquent, lors de la réutilisation de composants existants et du code source de jeux précédents, des tests robustes doivent être mis en œuvre.

Étapes d’implémentation

- Identifiez le code et les composants réutilisés : cataloguez le code source, les bibliothèques et les composants réutilisés dans les jeux précédents. Faites clairement la distinction entre le code maintenu activement et le code obsolète
- Documentez le comportement d'origine et les problèmes connus : enregistrez les caractéristiques de performance d'origine, les limitations fonctionnelles, les bogues connus ou les incidents de production associés aux composants réutilisés.
- Réalisez un examen complet du code : effectuez un examen technique détaillé des composants réutilisés, en particulier ceux qui ont rencontré des problèmes par le passé ou qui sont mal documentés.
- Remplacez ou refactorisez les composants existants à haut risque : privilégiez le remplacement ou la mise à jour des composants existants présentant des problèmes ou ne pouvant plus être maintenus, plutôt que de vous fier à des solutions de rechange en production.

- Effectuez des tests d'intégration et de compatibilité : validez les composants réutilisés dans le contexte des consoles du nouveau jeu. Vérifiez qu'ils interagissent correctement avec les nouveaux modules, outils et APIs.

GAMEOPS03-BP02 Procéder à l'ingénierie des performances avant chaque sortie (ou du moins pour les versions majeures)

L'ingénierie des performances est le processus qui consiste à surveiller plusieurs indicateurs opérationnels clés d'une application afin de découvrir des opportunités d'optimisation susceptibles d'améliorer encore les performances de l'application. Il s'agit d'un processus itératif qui commence par des tests, puis par l'optimisation du code, de ses dépendances, des processus associés, de son système d'exploitation hôte et de l'infrastructure sous-jacente.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Pour effectuer une analyse plus approfondie des performances de l'application, intégrez un outil de surveillance des performances de l'application (APM) ou de débogage dans le code de l'application qui permet d'isoler les problèmes et de réduire le temps de dépannage en suivant son comportement pour détecter les anomalies dans les flux de l'application. Les outils APM sont également capables d'identifier les méthodes lentes et les opérations externes.

AWS X-Rayaide les développeurs dans leurs activités d'ingénierie des performances, notamment en identifiant les goulots d'étranglement liés aux performances et en analysant et en corrigeant les erreurs de production. Vous pouvez utiliser X-Ray pour comprendre les performances de votre application et de ses services sous-jacents, ainsi que pour identifier et résoudre les causes profondes des problèmes et des erreurs de performance. Grâce à de nombreux tests de charge, au cours desquels l'application et son infrastructure sont progressivement chargées avec du trafic synthétique entre joueurs, divers goulets d'étranglement du système, des erreurs d'application, des exceptions, des problèmes de système d'exploitation et d'autres problèmes ont été identifiés qui n'auraient peut-être pas été détectés lors d'autres tests d'assurance qualité.

Pour les événements critiques tels que les lancements de jeux, les sorties de contenu, les promotions et les événements majeurs en jeu, utilisez AWS Countdown, qui fournit des conseils de mise en œuvre basés sur des playbooks élaborés par des experts du jeu pour vérifier l'état de préparation opérationnelle, atténuer les risques potentiels et planifier les besoins en capacités.AWS Countdown

propose également une option de support premium qui offre un support amélioré et des options telles que des ingénieurs pour optimiser votre infrastructure.

Étapes d’implémentation

- L'ingénierie des performances consiste à évaluer et à surveiller les indicateurs opérationnels clés afin de vérifier que le code, les processus, le système d'exploitation et l'infrastructure de votre application fonctionnent comme prévu. L'examen de pré-production permet également de définir les performances de référence à différents niveaux d'utilisation simulée.
- Découvrez et suivez les indicateurs clés tels que l'utilisation, les services, les E/S, les processus, etc. à l'aide d'outils système tels que sar, top, vmstat, sysstat, netstat et Performance Monitor.
- Suivez les performances et le comportement de votre application à l'aide d'outils APM, tels que AWS X-Ray pour isoler les problèmes, identifier les goulots d'étranglement et corriger les erreurs de production.
- Pour les événements critiques tels que le lancement de jeux, abonnez-vous à AWS Countdown (IEM) pour bénéficier de conseils architecturaux et opérationnels, d'un support opérationnel à la demande, ainsi que pour identifier les risques et planifier des mesures d'atténuation.

GAMEOPS03-BP03 Testez la charge tôt et souvent

Les tests de charge sont le processus qui consiste à simuler le trafic réel sur un système afin d'évaluer sa fiabilité et ses performances.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les tests de charge constituent un facteur clé pour développer une base de performance pour vos ressources et comprendre la capacité de votre système, ce qui peut orienter les prévisions financières, la conception de l'architecture, l'allocation des ressources, les configurations de dimensionnement automatisées et les activités de pré-dimensionnement après le lancement. Les avantages supplémentaires incluent :

- Infrastructure optimisée : les ressources peuvent être surprovisionnées ou sous-approvisionnées. La compréhension des ressources nécessaires se traduira par une baisse des coûts et une réduction de l'infrastructure à gérer.
- Préparation à l'évolutivité : certains mécanismes et fonctionnalités peuvent inciter les utilisateurs à se lancer rapidement dans un jeu. Savoir quand et comment évoluer peut faire la différence entre

répondre de manière appropriée à la demande accrue et perdre des joueurs. Utilisez les résultats des tests de charge pour préparer des runbooks avec des seuils système, des points d'alerte et des points d'alerte critiques à différents niveaux d'échelle.

- Code de meilleure qualité : les problèmes tels que la diaphonie excessive entre les services, les appels de base de données non groupés, les algorithmes inefficaces, les fuites de mémoire et les problèmes de dégradation des services sont parfois plus simples à identifier à grande échelle.
- Validation du comportement : l'injection de différents types d'échecs dans vos tests peut valider le comportement attendu du système ou révéler des problèmes de gestion des erreurs qui doivent être corrigés.

Idéalement, les développeurs devraient effectuer des tests de charge à plusieurs étapes du processus de développement, car chacun peut apporter des avantages différents : ils orientent les décisions architecturales et les efforts de refactorisation dès le début, alors que les modifications sont moins coûteuses et plus simples. À la fin de chaque sprint ou itération, ils valident les performances de l'application avec les dernières fonctionnalités.

Avant le déploiement en production, des tests de charge à grande échelle simulant les modèles d'utilisation attendus dans le monde réel confirment la capacité du système à gérer la charge de travail de production. Après le déploiement, des tests de charge périodiques surveillent les performances du système et identifient les modifications ou les blocages susceptibles de survenir au fil du temps.

Pour simuler le trafic des joueurs, vous avez besoin de clients légers ou de robots qui émulent les flux de clients du jeu et effectuent des transactions avec le backend du jeu pour simuler le comportement des joueurs dans le monde réel. Ces données sont généralement collectées par le biais de journaux de jeu et de données générées par des tests d'assurance qualité menés par des humains, ainsi que par le biais de tests alpha ou bêta à échelle limitée dans le monde réel au cours desquels de vrais joueurs sont invités à jouer à une version du jeu en accès anticipé.

Il est important d'enregistrer le comportement du système dans un manuel d'exploitation afin de faciliter le dépannage des défaillances éventuelles à l'avenir et de conserver les indicateurs de performance auxquels les futurs tests de charge pourront être comparés. Il est également recommandé de faire tester le jeu par un personnel d'assurance qualité humain pendant le test de charge, car il pourrait découvrir des problèmes que les robots ne parviennent pas à identifier et que les indicateurs ne reflètent pas.

AWS Le service d'injection de défauts est un service entièrement géré permettant d'exécuter des expériences d'injection de défauts qui facilitent l'amélioration des performances, de l'observabilité

et de la résilience d'une application. Les expériences d'injection de défauts sont utilisées dans l'ingénierie du chaos, qui consiste à stresser une application dans des environnements de test ou de production en créant des événements perturbateurs, tels qu'une augmentation soudaine de la consommation du processeur ou de la mémoire, en observant la réponse du système et en mettant en œuvre des améliorations. Les expériences d'injection de défauts aident les équipes à créer les conditions réelles nécessaires pour découvrir les bogues cachés, surveiller les angles morts et les obstacles aux performances difficiles à détecter dans les systèmes distribués.

Étapes d’implémentation

- Configurez un environnement de test de charge distribué à l'aide de Guidance for Kubernetes-Bases Game Load Testing.
- Personnalisez et déployez des modules de contrôle et de travail Locust au sein du cluster EKS à l'aide des fichiers de déploiement fournis, ce qui permet une génération de charge évolutive et gérable.
- Enregistrez le comportement et les indicateurs du système lors des tests de charge dans un manuel opérationnel afin de faciliter le dépannage futur et d'établir des références de performance.
- Utilisez des expériences d'injection de défauts pour simuler des perturbations réelles et découvrir les problèmes cachés liés aux performances, à l'observabilité et à la résilience du système.

GAMEOPS03-BP04 Adopter une stratégie de déploiement qui minimise l'impact sur les joueurs

Intégrez une stratégie de déploiement pour votre logiciel de jeu et votre infrastructure afin de minimiser les temps d'arrêt qui empêchent les joueurs de jouer. Certains types de mises à jour peuvent nécessiter l'installation de nouvelles mises à jour sur le client du jeu, mais concevez le jeu de manière à minimiser ou à éviter les interruptions de service lors des déploiements.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

L'une des étapes les plus importantes à prendre en compte lors de l'élaboration d'une stratégie de déploiement de jeu est de déterminer comment votre infrastructure de jeu sera gérée. Gérez votre infrastructure de jeu à l'aide d'un outil d'infrastructure en tant que code (IaC) tel que AWS CloudFormationTerraform de Hashicorp afin de réduire les erreurs humaines lors de la préparation de l'environnement. Les modèles d'infrastructure peuvent être déployés et testés dans des pipelines

automatisés, ce qui assure une cohérence dans la configuration des différents environnements de jeu.

Plusieurs stratégies de déploiement peuvent être utilisées pour un jeu :

Substitution par roulement

L'objectif principal d'une substitution progressive pour le déploiement est d'effectuer la sortie sans arrêter le jeu et sans affecter les joueurs. Il est important que la mise à niveau ou les modifications à effectuer soient rétrocompatibles et fonctionnent de manière similaire aux versions précédentes du système.

Dans ce déploiement, les instances de serveur sont progressivement remplacées (substituées ou déployées) par des instances exécutant la version mise à jour. Cette substitution par roulement peut être réalisée de différentes manières. Par exemple, pour implémenter des mises à jour continues sur un parc de serveurs de jeu dédiés, une approche classique consiste à créer un nouveau groupe d' EC2 instances Auto Scaling contenant la nouvelle version de build du serveur de jeu déployée sur celles-ci, puis à acheminer progressivement les joueurs vers des sessions de jeu hébergées sur ce nouveau parc de serveurs. Si une mise à jour du client de jeu associée est requise comme condition préalable à l'utilisation de la nouvelle version du serveur de jeu, vous devez inclure un contrôle de validation pour vérifier que seuls les joueurs sur lesquels cette nouvelle mise à jour du client de jeu est installée sont redirigés vers ces sessions de jeu.

Les flottes de serveurs (par exemple, les groupes EC2 Auto Scaling) contenant l'ancienne version du serveur de jeu ne sont retirées du service qu'après avoir été vidées des sessions de joueurs actives de manière progressive, généralement en configurant des métriques de serveur individualisées qui permettent aux équipes chargées des opérations de jeu d'automatiser ce processus. Pour réduire la quantité d'infrastructure et le temps nécessaires à un déploiement continu, il est également possible d'adopter une autre approche consistant à retirer les instances de production existantes du service, à les mettre à jour avec la nouvelle version du serveur de jeu, puis à les réintégrer dans le parc de production. Cette approche réduit la quantité d'infrastructure requise, mais elle augmente également les risques, car le nombre de serveurs de jeu en direct disponibles pour les joueurs diminue à mesure que les serveurs sont remplacés.

Ce modèle peut également être utilisé pour effectuer des déploiements progressifs sur des services principaux tels que les bases de données, les caches et les serveurs d'applications qui n'hébergent pas le gameplay. Tant que ces services sont déployés de manière hautement disponible avec plusieurs instances en cluster, la complexité des déploiements vers ces services devrait être moindre que celle des déploiements sur des serveurs de jeu dédiés.

Déploiement bleu/vert

L'objectif principal d'un blue/green déploiement dans un jeu est de minimiser les temps d'arrêt tout en permettant de revenir en toute sécurité au déploiement précédent si des problèmes sont identifiés. Il convient aux déploiements où deux versions du backend du jeu sont compatibles et peuvent servir les joueurs simultanément.

Dans la stratégie de blue/green déploiement, deux environnements identiques (bleu et vert) sont configurés. La version de jeu existante est étiquetée en bleu, tandis que la nouvelle version du jeu cible de déploiement est étiquetée en vert. Lorsque l'environnement vert est prêt pour la migration, vous pouvez configurer votre couche de routage pour transférer le trafic vers l'environnement vert tout en gardant l'ancien environnement (bleu) disponible au cas où un retour en arrière serait nécessaire. Dans ce scénario, les mises à jour du routage peuvent nécessiter la mise à jour du service de matchmaking afin de le configurer afin de commencer à envoyer des sessions de jeu à la nouvelle flotte, ou dans le cas des services de backend de jeu, il peut s'agir de mettre à jour les enregistrements DNS dans Amazon Route 53 pour votre service ou de modifier les pondérations de l'équilibreur de charge des applications pour envoyer le trafic vers votre nouveau groupe cible.

L'un des inconvénients de la stratégie de blue/green déploiement est le coût inhérent à l'environnement de veille en raison de l'infrastructure supplémentaire requise pour effectuer le déploiement. Une option pour atténuer ce coût d'infrastructure supplémentaire consiste à envisager d'adopter une variante de blue/green déploiement dans laquelle les nouveaux logiciels de jeu sont déployés sur les mêmes serveurs que ceux déjà déployés en production. Dans ce scénario, un nouveau processus de serveur vert peut être lancé avec le nouveau logiciel parallèlement au processus de serveur bleu existant, le passage se faisant entre les processus du serveur plutôt qu'entre des infrastructures physiques distinctes. Cette approche peut également accélérer les déploiements de jeux sur une grande partie de l'infrastructure en éliminant le besoin d'attendre le lancement de nouveaux serveurs dans le cloud. Pour connaître les meilleures pratiques relatives à cette approche de déploiement, voir Déploiements bleu/vert sur. AWS

Déploiement Canary

Le déploiement de Canary est utile pour les développeurs de jeux, car la stratégie peut être appliquée pour publier une première version alpha ou bêta d'un jeu, ou une fonctionnalité de jeu telle qu'un nouveau mode de jeu, une nouvelle carte ou un défi lancé à un groupe restreint ou restreint de joueurs en production. Un tel déploiement s'appelle un canari. La version peut comporter un suivi et des rapports supplémentaires. Ainsi, lorsque de vrais joueurs jouent à ce jeu ou à cette fonctionnalité, leur télémétrie de jeu est collectée et analysée pour détecter les anomalies et les problèmes.

En ce qui concerne les nouvelles fonctionnalités, les joueurs n'en sont pas systématiquement informés, et la télémétrie du jeu est la principale source utilisée pour déterminer si les joueurs rencontrent des problèmes et si la version doit être annulée. Dans le même temps, si aucun problème significatif n'est identifié, la fonctionnalité peut ensuite être déployée auprès d'un plus grand nombre de joueurs pour obtenir des données supplémentaires. Si les joueurs sont informés, ils peuvent être invités à fournir régulièrement des commentaires sur leur expérience. Une telle activité de test devrait idéalement être coordonnée par une équipe opérationnelle en direct.

En tant que stratégie, le déploiement de Canary peut également être utilisé pour les versions standard afin de mettre progressivement une nouvelle fonctionnalité à la disposition des joueurs. L'un des avantages potentiels par rapport à blue/green l'environnement standard est qu'un deuxième environnement complet n'est pas nécessaire. La capacité du nouvel environnement réduit détermine le nombre de joueurs à intégrer à la nouvelle fonctionnalité. Avant d'ajouter d'autres joueurs, la capacité doit être adaptée de manière appropriée. Même si cette blue/green technique personnalisée est censée coûter relativement moins cher que le bleu/vert standard, on estime qu'elle entraîne des coûts qui peuvent être supérieurs à ceux de la technique de substitution progressive des déploiements Canary.

N'exécutez qu'un seul Canary dans un environnement de production et concentrez-le sur ses données et ses commentaires. Si plusieurs canaris sont déployés, cela complique le dépannage et l'identification des problèmes de production et nuit à la qualité des ensembles de données et des commentaires collectés.

Une variante du canari se produit lorsqu'une ou plusieurs expériences (généralement des tests d'interface utilisateur) sont menées dans le cadre de déploiements ciblés, où un ensemble de serveurs principaux du jeu propose une version d'une fonctionnalité et un autre ensemble de même taille propose une autre version de la même fonctionnalité. Aucune infrastructure supplémentaire ou spéciale n'est créée pour cela, et seuls les groupes de serveurs principaux sélectionnés reçoivent ces mises à jour. Le résultat des expériences est d'observer la façon dont les joueurs réagissent à chacune des versions d'une même fonctionnalité, de déterminer s'il existe un consensus général sur le point d'aimer ou de ne pas aimer, et d'observer si des problèmes d'utilisabilité ou de fonctionnalité ont été identifiés. Ces expériences stratégiques sont également appelées A/B tests, et le processus global est appelé test A/B. À la fin de ces expériences, les données de test nécessaires sont collectées avant de revenir à la version actuelle du système principal du jeu sur les serveurs utilisés pour les tests.

Déploiements traditionnels traditionnels

Dans le style de déploiement traditionnel, pendant une période de maintenance planifiée, le jeu est arrêté et les joueurs connectés sont supprimés ou vidés avant que les instances de serveur du backend du jeu ne soient mises à jour avec les dernières versions de code. Ce déploiement a un impact sur les joueurs à chaque fois qu'il est effectué, et les joueurs doivent être prévenus à l'avance. Par conséquent, ce modèle a le plus d'impact sur les joueurs et doit être évité dans la mesure du possible.

Une fois la mise à jour déployée, le jeu peut être testé avant d'être ouvert aux joueurs, qui attendront sa réouverture. Cela peut provoquer un pic de trafic lorsque les joueurs essaient de se connecter et de jouer dans un court laps de temps. Par conséquent, si le jeu n'est pas conçu pour gérer de tels pics de trafic, vous pouvez choisir d'autoriser progressivement les joueurs à revenir dans le jeu par lots.

Vous pouvez également choisir de surapprovisionner l'infrastructure pour faire face au pic de trafic initial, et une fois le trafic du jeu réglé, les ressources peuvent être réduites. Si nécessaire, effectuez ce type de déploiement en dehors des heures de pointe, lorsque le nombre de joueurs est au plus bas. Les maintenances fréquemment programmées, ainsi que les maintenances prolongées, comportent intrinsèquement un risque d'attrition des joueurs et de pertes de revenus potentielles. Les joueurs s'attendent également à des changements après une nouvelle version et peuvent perdre confiance dans le jeu à leur retour après une période d'indisponibilité.

Étapes d’implémentation

- Minimisez les temps d'arrêt : mettez en œuvre des stratégies de déploiement qui réduisent les temps d'arrêt et permettent aux joueurs de rester actifs.
- Infrastructure en tant que code (IaC) : utilisez des outils tels que AWS CloudFormation Terraform pour gérer l'infrastructure du jeu et réduire les erreurs humaines.
- Stratégies de déploiement : utilisez un ou plusieurs des déploiements par substitution progressive, bleu/vert et Canary pour garantir des mises à jour fluides et réduire l'impact sur les joueurs.

GAMEOPS03-BP05 Infrastructure prédimensionnée requise pour répondre aux exigences de pointe

Développez votre infrastructure avant les événements de jeu de grande envergure afin de pouvoir faire face à l'augmentation soudaine de la demande de joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Outre les lancements de nouveaux jeux, les jeux en direct proposent généralement des événements intégrés au jeu, des promotions, du nouveau contenu et des sorties de saison comme exemples de moyens de maintenir et d'améliorer l'engagement des joueurs. Ces activités génèrent un volume élevé de trafic de joueurs pendant toute la durée de l'événement ou de la promotion. L'entreprise s'attend à atteindre ou à dépasser ses objectifs pour l'événement, et l'infrastructure du jeu doit les maintenir et les soutenir tout au long de cet événement.

Préparez votre infrastructure à l'avance afin de pouvoir supporter la charge de joueurs prévue lors d'événements de grande envergure. Pour se préparer, les équipes chargées des opérations de jeu doivent se coordonner avec les parties prenantes des ventes et du marketing afin d'estimer la demande prévue qui sera générée lors d'un prochain événement en examinant la simultanéité des joueurs passés, les indicateurs d'engagement et les données de vente. Si l'événement concerne le lancement d'un nouveau jeu, les équipes chargées des opérations de jeu doivent travailler avec ces parties prenantes pour identifier des projections réalistes quant à l'ampleur qu'elles anticipent. Bien qu'il soit difficile de prévoir le succès d'un jeu, il est important que chacun comprenne quelles sont les attentes en matière de succès afin que l'infrastructure puisse être adaptée et testée pour atteindre ces objectifs.

De nombreux jeux choisissent de se lancer par étapes, en commençant par un lancement progressif en ouvrant le jeu à un petit nombre de joueurs, puis en faisant évoluer le nombre de joueurs de manière organique à chaque étape, avant un lancement public complet. Pendant la période de lancement progressif, surveillez, identifiez, suivez et résolvez les problèmes tout en affinant vos prévisions pour le lancement public.

Pour évaluer correctement les besoins en infrastructure, collectez des données par le biais de tests de charge et de performance exécutés sur les backends de votre jeu en cours de production ou dans un environnement de mise en scène similaire à la production avant le lancement du jeu. Plusieurs séries de ces tests doivent être effectuées pour simuler les différentes conditions du jeu et valider que le backend peut supporter la charge dans la plupart des conditions.

Pour ce faire, les développeurs peuvent créer des robots de jeu qui parcourent les différents flux de travail du jeu et émulent différentes conditions. Ces tests doivent inspecter les différentes couches du système du backend du jeu afin que chaque couche et chaque composant soient testés et que les détails soient enregistrés. Utilisez les données collectées lors de ces tests pour planifier le lancement du jeu.

Les points de défaillance uniques (SPOF) doivent être identifiés et supprimés dans la mesure du possible en rendant l'application hautement disponible et tolérante aux pannes. Utilisez des tests de charge pour identifier SPOFs en imitant les défaillances sur différentes couches en amont et en aval et en vérifiant le comportement du jeu et des autres composants.

Outre l'infrastructure estimée nécessaire à fournir pour le lancement du jeu, les événements en jeu ou les préparatifs de promotion, configurez le système pour qu'il évolue automatiquement à la demande. Définissez, configurez et surveillez les seuils d'événements de dimensionnement pour permettre au backend du jeu de s'adapter à un volume élevé de trafic de joueurs. Pour le trafic variable, le préprovisionnement est préférable, car il se peut que le temps ne soit pas suffisant pour effectuer le scale-out. Une mise à l'échelle manuelle peut être nécessaire lors des premiers lancements de jeux, qui génèrent une demande plus élevée que prévu plus rapidement que les systèmes automatisés ne peuvent augmenter les ressources.

Oui AWS, les entreprises doivent demander des Quotas de Service plus élevés pour les services qu'elles utilisent dans le backend du jeu. Les Quotas de Service sont définis pour les comptes afin d'empêcher les clients d'installer ou de développer par inadvertance une infrastructure plus importante que prévu. Lorsqu'un jeu exécuté sur un compte atteint la limite supérieure du quota de service configuré dans cette région, le service limite les demandes au-delà du quota alloué et des provisions de rafale. Les accélérateurs peuvent provoquer des erreurs involontaires ou inattendues et nuire à l'expérience du joueur. Surveillez, suivez et révisez régulièrement les seuils de quotas de service pour les services utilisés par le jeu en production afin d'éviter les ralentissements. Lorsque l'utilisation dépasse un seuil de quota de service acceptable, une augmentation du quota peut être demandée en soumettant un dossier de support auprès du Centre de support de la console, après vous être connecté au compte concerné ou en utilisant l'API de support.

Pour les événements critiques tels que les lancements de jeux, les sorties de contenu, les promotions et les événements majeurs du jeu, utilisez AWS Countdown. Countdown fournit des conseils de mise en œuvre basés sur des playbooks élaborés par des experts des Jeux pour assurer la préparation opérationnelle, atténuer les risques potentiels et planifier les besoins en capacités.AWS Countdown propose également une option de support premium qui offre un support amélioré et des options telles que des ingénieurs pour optimiser votre infrastructure.

Si vous lancez un jeu hébergé sur Amazon GameLift, consultez les listes de contrôle avant le lancement pour vous préparer.

Étapes d’implémentation

- Développez l'infrastructure à l'avance : préparez l'infrastructure à l'avance pour les événements de jeu de grande envergure afin de faire face à l'augmentation soudaine de la demande des joueurs.
- Estimation de la demande : coordonnez-vous avec les équipes des ventes et du marketing pour estimer la demande prévue à l'aide des données des anciens joueurs et de projections réalistes.
- Tests de charge et suppression du SPOF : effectuez plusieurs séries de tests de charge pour valider la capacité du backend, identifier les points de défaillance uniques et configurer correctement le dimensionnement automatique.

Surveillance de la santé

GAMEOPS04 : Comment surveillez-vous l'état du jeu ?

Surveillez l'état du jeu en mettant en œuvre une instrumentation complète pour détecter et suivre les problèmes ayant un impact sur les joueurs, notamment la journalisation des activités côté client, la surveillance des services principaux et le signalement des erreurs. Utilisez une combinaison d' AWS outils tels qu'Amazon CloudWatch et AWS X-Ray des solutions tierces pour identifier et résoudre rapidement les problèmes.

Bonnes pratiques

- GAMESOPS04-BP01 Instrumenter le jeu pour détecter et surveiller les problèmes ayant un impact sur le joueur

GAMESOPS04-BP01 Instrumenter le jeu pour détecter et surveiller les problèmes ayant un impact sur le joueur

En plus de répondre aux signalements de problèmes sur les réseaux sociaux et aux signalements de problèmes par les joueurs, dotez votre jeu de solutions de surveillance permettant de détecter et d'étudier les problèmes ayant un impact sur les joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Aucun test ne permet d'identifier tous les problèmes d'un jeu. Les jeux sont généralement lancés avec des problèmes connus qui devraient être progressivement corrigés lors de la prochaine sortie du jeu. Les problèmes connus et reproductibles sont faciles à traiter et à corriger. Pour aider à identifier ces problèmes, les clients du jeu doivent implémenter le suivi de l'activité des joueurs, la journalisation des applications et le reporting à divers endroits stratégiques afin d'aider l'équipe principale à identifier les problèmes côté client. La capacité de détecter ces problèmes à un stade précoce aide les développeurs de jeux à les résoudre et à les résoudre avant qu'ils ne se généralisent. Les données et les journaux signalés par le code de suivi ne doivent jamais inclure d'informations personnelles identifiables (PII), et ils ne doivent contenir que des métadonnées spécifiques au jeu qui facilitent le débogage.

Mettez en œuvre une solution d'observabilité pour détecter et résoudre les problèmes tels que les plantages de jeu ou les bogues. Vous pouvez utiliser Amazon CloudWatch Synthetics pour créer des canaris capables de surveiller l'état de vos services de jeu principaux destinés aux joueurs. Vous pouvez utiliser les instruments de vos services principaux AWS X-Raypour suivre les demandes sur l'ensemble des services distribués et envoyer vos journaux et indicateurs personnalisés à Amazon CloudWatch.

Les solutions tierces, telles que Backtrace.io et Sentry, sont des solutions populaires pour le signalement d'erreurs dans les jeux. Les solutions de surveillance des performances des applications (APM) proposées par des partenaires tels que New Relic, Splunk, Datadog et Honeycomb.io sont également populaires.

L'équipe des opérations en direct du jeu et les responsables de communauté devraient également surveiller les différents réseaux sociaux et canaux pour vérifier les commentaires des joueurs, les plaintes et les rapports de bogues, en plus des canaux d'assistance officiels. Passez en revue et essayez de reproduire chaque plainte spécifique au jeu, ou envoyez-la à l'équipe d'assurance qualité pour examen. S'il est reproductible, signalez le problème aux développeurs du jeu pour qu'ils le résolvent et qu'ils trouvent une solution avant qu'il n'ait un impact sur l'ensemble de la base de joueurs.

Étapes d’implémentation

- Mettez en œuvre des solutions de surveillance : utilisez des outils de surveillance pour détecter les problèmes ayant un impact sur les joueurs et réagir rapidement.

- Suivez l'activité et les journaux des joueurs : les clients du jeu d'instruments enregistrent l'activité des joueurs, signalent les problèmes et vérifient qu'aucune information personnelle identifiable (PII) n'est incluse.
- Utilisez des AWS outils et des outils tiers : utilisez des outils tels que CloudWatch X-Ray et des solutions tierces pour signaler les erreurs et surveiller les performances, et surveillez les réseaux sociaux pour connaître les commentaires des joueurs et les rapports de bogues.

Test de charge

GAMEOPS05 : Que devez-vous prendre en compte lorsque vous testez le chargement d'un jeu ?

Lorsque vous testez la charge d'un jeu, considérez l'étape de test, l'architecture génératrice de charge et le cadre de test appropriés pour évaluer efficacement les performances et l'évolutivité du système. Choisissez la bonne combinaison de timing (développement initial, sprints, pré-production ou post-déploiement), d'infrastructure (EKSEC2, Fargate ou Lambda) et d'outil de test (JMeterLocust, Grafana K6 ou Gatling) pour correspondre aux caractéristiques uniques et aux objectifs de développement de votre jeu.

Bonnes pratiques

- GAMEOPS05-BP01 Choisissez le stage, l'architecture et le framework de test de charge adaptés à vos objectifs

GAMEOPS05-BP01 Choisissez le stage, l'architecture et le framework de test de charge adaptés à vos objectifs

L'approche utilisée pour tester la charge d'un jeu peut varier considérablement en fonction de nombreux facteurs, notamment le stade du processus de développement dans lequel il est réalisé, l'architecture du système de génération de charge lui-même et le choix du framework de test de charge. Le moment où il est effectué, que ce soit dans les premières phases, pendant les sprints itératifs, avant le déploiement en production ou après le déploiement, déterminera les objectifs et l'orientation des efforts de test. Les différentes conceptions d'infrastructure génératrice de charge ont leurs avantages et leurs inconvénients, et le choix du framework de test de charge influence considérablement les capacités, la facilité d'utilisation et les intégrations disponibles pour le processus de test. En alignant judicieusement ces éléments, les équipes de développement peuvent

adapter l'approche des tests de charge aux caractéristiques uniques du jeu, extraire les informations les plus précieuses sur les performances et offrir une expérience fluide à leurs joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Tests de charge à différents stades de développement

La réalisation de tests de charge exploratoires au début des phases de développement permet de valider l'architecture du système sous-jacent. Cela permet aux développeurs de prendre des décisions éclairées concernant l'infrastructure du jeu, la conception de la base de données et la topologie du réseau avant d'effectuer un travail de mise en œuvre approfondi. Les tests de charge identifient les risques et créent une référence de performance, ce qui permet de minimiser le besoin de retouches coûteuses et de dettes techniques plus tard dans le cycle de développement. Ils peuvent également favoriser une compréhension partagée des exigences de performance du jeu au sein de l'équipe, ce qui se traduit par une meilleure collaboration et une meilleure prise de décision. En fin de compte, les tests de charge effectués au cours des phases initiales constituent une base solide pour un jeu performant, évolutif et résilient, contribuant ainsi à améliorer l'expérience globale des joueurs.

À la fin de chaque sprint ou itération, les tests de charge permettent d'évaluer l'impact sur les performances des nouvelles fonctionnalités, des corrections de bogues et des autres modifications introduites au cours du dernier cycle. Cette approche ciblée permet aux équipes de développement d'identifier rapidement les régressions ou les dégradations de performances introduites par les dernières mises à jour, ce qui leur permet de résoudre ces problèmes avant qu'ils ne se propagent plus loin dans le pipeline et de maintenir un niveau constant de qualité et de performance.

Avant le déploiement en production, des tests de charge robustes aident les équipes à valider la capacité du système à gérer les conditions de trafic et de charge réelles prévues. Ils peuvent identifier les goulots d'étranglement liés à l'évolutivité ou les contraintes de ressources au sein de l'infrastructure de production et offrir la possibilité d'optimiser les performances du jeu, en créant une expérience utilisateur fluide et réactive dès le premier jour. Les informations recueillies lors des tests de charge effectués avant le lancement peuvent atténuer les risques liés au lancement et éclairer la planification continue des capacités, qui jette les bases de la durabilité et de l'évolutivité à long terme du jeu.

Les tests de charge d'un jeu déjà en production permettent aux équipes de surveiller les performances du jeu et d'identifier les régressions ou les dégradations de performances susceptibles

de se produire au fil du temps. Cela leur permet de résoudre les problèmes de manière proactive avant qu'ils n'affectent l'expérience des joueurs et n'affectent négativement la fidélisation des utilisateurs. En outre, les tests de charge en production valident l'efficacité des efforts d'optimisation des performances ou de mise à l'échelle de l'infrastructure mis en œuvre. Ce processus fournit aux joueurs une expérience de jeu de haute qualité, réactive et évolutive, même au fur et à mesure que le jeu évolue et mûrit.

Architectures génératrices de charge

La conception de l'architecture de génération de charge pour les tests de chargement des jeux peut prendre différentes formes, chacune présentant ses propres avantages et considérations.

Au niveau le plus élémentaire, les EC2 instances Amazon autogérées peuvent être mises en service et configurées pour agir comme des générateurs de charge. Grâce à l'approche des nœuds de contrôle et des nœuds de travail, vous pouvez configurer plusieurs instances génératrices de charge, chacune exécutant son propre script de test et étant globalement gérée par une seule instance de contrôle. L'architecture peut évoluer et générer plus de charge sans augmenter la complexité en installant des nœuds de travail supplémentaires, mais cette approche pratique oblige les équipes à gérer le provisionnement, la configuration et la gestion de l'infrastructure sous-jacente.

Pour une approche plus évolutive et orchestrée, vous pouvez utiliser les clusters Amazon EKS Kubernetes pour gérer et répartir la charge de travail des tests de charge sur un parc d'agents de charge basés sur des conteneurs. Les fonctionnalités de dimensionnement automatique de Kubernetes peuvent être utilisées pour gérer le dimensionnement des pods générateurs de charge, tandis que les équipes configurent et gèrent elles-mêmes les EC2 instances sous-jacentes du cluster hébergeant les pods.

La nature sans serveur de AWS Fargatepeut également accélérer et simplifier la configuration des tests de charge en supprimant la gestion de l'infrastructure tout en offrant l'évolutivité et la flexibilité nécessaires. Pour les solutions hybrides dans lesquelles un cluster Kubernetes sur site générateur de charge existe déjà mais qu'une capacité supplémentaire peut être nécessaire, EKS Anywhere peut gérer les deux clusters comme s'il s'agissait d'un seul cluster à partir du. AWS Management Console

Vous pouvez également utiliser des AWS Lambdafonctions en fonction de vos besoins et de vos objectifs. Les fonctions Lambda sont relativement simples à configurer et à faire évoluer sans qu'il soit nécessaire de fournir et de gérer des ressources supplémentaires. Ils permettent également de créer des scénarios de test plus complexes et dynamiques grâce à une intégration approfondie avec d'autres AWS services. Cependant, les fonctions Lambda sont soumises à des limites quant aux fonctions simultanées et à leur durée d'exécution (15 minutes), ce qui peut limiter l'échelle et la durée

des tests de charge pouvant être réalisés. Les latences de démarrage à froid peuvent également avoir un impact sur la précision des résultats, et les limites de ressources de Lambda peuvent ne pas être adaptées aux charges de travail de test de charge très exigeantes.

Les studios qui souhaitent utiliser une solution prédéfinie peuvent utiliser les tests de charge distribués sur AWS. Cette solution utilise Amazon ECS AWS Fargate pour déployer des conteneurs capables d'exécuter des simulations de dizaines de milliers d'utilisateurs connectés. Vous pouvez l'utiliser pour démarrer rapidement votre infrastructure de test de charge à la manière IAC en utilisant AWS CloudFormation.

Cadres de test de charge

Il n'existe pas deux frameworks de test de charge conçus de la même manière. Certains disposent d'interfaces graphiques intuitives pour la création de tests, tandis que d'autres sont entièrement basés sur la ligne de commande. Un outil peut être flexible et performant, mais sa configuration et sa gestion nécessitent du temps et des efforts, tandis qu'un autre peut être sans serveur mais limité dans les tests qu'il peut créer et exécuter. Certains bénéficient de vastes communautés et de nombreux tutoriels alors qu'ils n'ont pas encore fait leurs preuves sur le terrain, ce qui contraste nettement avec d'autres qui peuvent avoir fait leurs preuves en production mais qui manquent de soutien ou de documentation de la part de la communauté. Choisissez le cadre qui offre le bon équilibre pour vous et votre équipe. Voici quelques options populaires :

- Apache JMeter : framework de test de charge open source populaire basé sur Java en raison de ses fonctionnalités robustes et de sa facilité d'utilisation. Sa capacité à simuler des scénarios utilisateur complexes, son large éventail de protocoles pris en charge, ses rapports complets et ses antécédents éprouvés en font JMeter un choix fiable pour les tests de charge.
- Locust : Framework de test de charge moderne et distribué basé sur une architecture axée sur les événements, ce qui le rend performant tout en économisant les ressources. Les tests sont écrits en Python, ce qui permet des scénarios de test flexibles qui tirent parti de milliers de puissantes bibliothèques tierces, tout en restant conviviaux et simples à lire.
- Grafana K6 : puissant framework de test de charge alliant facilité d'utilisation et fonctionnalités avancées. Sa prise en charge de la génération de charge distribuée, ses scripts flexibles et son intégration parfaite avec Grafana pour la visualisation des données font de Grafana K6 un choix intéressant.
- Gatling : framework de test de charge open source connu pour ses performances et son évolutivité. Son langage spécifique au domaine (DSL) basé sur Scala permet aux développeurs de créer des scripts de test de charge concis et faciles à gérer, et ses fonctionnalités robustes de reporting et d'analyse fournissent des informations détaillées sur le système testé.

Étapes d’implémentation

- Étapes de test de charge : effectuez des tests de charge à différentes étapes de développement (développement initial, sprints, pré-production et post-déploiement) pour valider les performances du système et identifier les problèmes.
- Architectures génératrices de charge : choisissez les architectures de génération de charge appropriées (EC2EKS, Fargate ou Lambda) en fonction des besoins d'évolutivité, des préférences de gestion et des exigences de test spécifiques.
- Frameworks de test de charge : sélectionnez un framework de test de charge (comme JMeter Locust, Grafana K6 ou Gatling) qui concilie facilité d'utilisation, performance, flexibilité et soutien communautaire pour répondre aux besoins de votre équipe.

Optimisation au fil du temps

GAMEOPS06 : Comment optimisez-vous votre jeu au fil du temps ?

Optimisez votre jeu au fil du temps en surveillant les indicateurs clés et les données de télémétrie afin d'identifier les tendances des joueurs, les performances du système et les points à améliorer. Mettez continuellement à jour la conception de votre jeu, votre infrastructure et votre approche des tests de charge en fonction de ces informations, tout en vous adaptant aux nouvelles technologies et frameworks afin d'offrir des performances et une expérience de jeu optimales au fur et à mesure de l'évolution de votre jeu.

Bonnes pratiques

- GAMEOPS06-BP01 Surveillez les indicateurs clés du jeu pour identifier les tendances et les habitudes des joueurs, et utilisez les informations pour améliorer le jeu
- GAMEOPS06-BP02 Mettre à jour et adapter l'approche des tests de charge à mesure que le jeu évolue

GAMEOPS06-BP01 Surveillez les indicateurs clés du jeu pour identifier les tendances et les habitudes des joueurs, et utilisez les informations pour améliorer le jeu

Outre l'utilisation du système client du jeu, l'utilisation des applications, les exceptions et les données de crash, capturez les données de télémétrie du jeu qui sont envoyées à un système principal de jeu. Ces données doivent représenter l'activité des joueurs afin que vous puissiez comprendre comment les joueurs interagissent avec les différentes fonctionnalités du jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

En fonction de sa mise en œuvre, les clients du jeu peuvent collecter des données de télémétrie à des fonctionnalités de jeu ou à des emplacements prédéfinis dans un monde de jeu. Les données sont envoyées au service d'ingestion principal pour traitement. Si le service principal est inaccessible, les clients peuvent stocker les données localement sur l'appareil local jusqu'à ce que le service principal soit à nouveau disponible. Les concepteurs de jeux utilisent ces données de télémétrie pour examiner la façon dont les joueurs jouent au jeu et s'il y a des anomalies dans le jeu.

Par exemple, les mouvements des joueurs et leurs interactions avec les éléments d'une carte peuvent être extraits des données de télémétrie et tracés sous forme de carte des activités des joueurs en jeu sur une période définie. Ces données aident les concepteurs de jeux à identifier la nécessité d'équilibrer les différents éléments du jeu, tels que la puissance d'une arme, la puissance d'un personnage du jeu ou la complexité d'une carte. Les données de télémétrie brutes sont généralement stockées puis traitées pour extraire des analyses qui peuvent être visualisées par les analystes.

La mise en œuvre de la AnalyticsPipeline solution Game aide les développeurs de jeux à lancer un pipeline de données évolutif sans serveur pour ingérer, stocker et analyser les données de télémétrie générées par les jeux et les services. La solution prend en charge l'ingestion de données en streaming, permettant aux utilisateurs d'obtenir des informations sur leurs jeux et autres applications en quelques minutes.

Pour la télémétrie personnalisée des jeux, l'ingestion, le stockage, le traitement et l'analyse de données, propose AWS également un certain nombre de services spécialisés pour le traitement des mégadonnées et l'analyse.

Étapes d’implémentation

- Capturez les données de télémétrie du jeu : collectez des données sur l'activité des joueurs, l'utilisation du système, les exceptions et les pannes afin de comprendre les interactions des joueurs et d'identifier les problèmes.
- Mettre en œuvre la collecte de données télémétriques : utilisez des fonctionnalités ou des emplacements de jeu prédéfinis pour collecter des données de télémétrie et les envoyer aux services principaux, en les stockant localement si le backend est inaccessible.
- Utilisez des solutions AWS d'analyse : utilisez AWS des services tels que le Game Analytics Pipeline pour l'ingestion, le stockage et l'analyse de données évolutifs, ainsi que des services spécialisés de traitement et d'analyse des mégadonnées.

GAMEOPS06-BP02 Mettre à jour et adapter l'approche des tests de charge à mesure que le jeu évolue

L'optimisation de l'approche des tests de charge est un processus continu qui devrait évoluer parallèlement au cycle de développement du jeu. À mesure que le jeu gagne en complexité, en base d'utilisateurs et en fonctionnalités, la stratégie de test de charge doit s'adapter pour vérifier qu'elle simule avec précision les conditions réelles et fournit des informations exploitables.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Éléments à prendre en compte :

Scénarios de test manquants ou obsolètes

Au fur et à mesure que de nouvelles fonctionnalités sont ajoutées à un jeu au cours du processus de développement, créez et exécutez de nouveaux scénarios de test de charge pour valider les performances et l'évolutivité des nouvelles fonctionnalités. De même, les caractéristiques et fonctionnalités sont souvent remaniées pour améliorer les performances, répondre aux commentaires des joueurs ou s'aligner sur les nouveaux objectifs de conception, ce qui nécessite la mise à jour continue des scénarios de test pour suivre le rythme des changements et tester et refléter véritablement l'état du système.

Nouveaux cadres de test de charge

Les développeurs peuvent avoir besoin de modifier les frameworks de test de charge pour diverses raisons :

- Le framework initial peut ne plus être en mesure de simuler correctement la charge utilisateur ou de fournir le niveau d'information nécessaire sur les performances du système
- Les nouvelles fonctionnalités du jeu peuvent nécessiter la prise en charge de tests de charge pour les APIs nouveaux protocoles ou points d'intégration
- Les développeurs souhaiteront peut-être des fonctionnalités plus avancées à mesure qu'ils se familiariseront avec le processus de test de charge
- Préférence pour les frameworks qui correspondent mieux à l'expertise technique de l'équipe, aux langages de programmation ou aux chaînes d'outils existantes

En évaluant soigneusement et en s'adaptant au fil du temps, les développeurs peuvent adapter le processus de test de charge aux exigences changeantes du jeu et continuer à fournir les informations nécessaires pour optimiser et améliorer l'expérience utilisateur globale.

Optimisation des coûts

La facilité et la commodité d'utilisation AWS des services gérés peuvent être très bénéfiques, en particulier au début du développement. Ces services font abstraction de la gestion sous-jacente de l'infrastructure, permettant aux équipes de configurer rapidement leur solution et de se concentrer uniquement sur l'élaboration de scénarios de test de charge et l'analyse des résultats. Cependant, l'utilisation de services gérés peut souvent coûter plus cher en raison de la valeur ajoutée et de la commodité qu'ils apportent, comme le provisionnement, la configuration et la maintenance de l'infrastructure, ainsi que la fourniture de fonctionnalités de haute disponibilité, d'évolutivité et de surveillance.

À mesure que les équipes mûrissent et deviennent plus à l'aise et confiantes dans leur processus de test de charge, il peut arriver un moment où l'autogestion de l'infrastructure peut apporter une optimisation supplémentaire et des économies de coûts. Bien que cette approche pratique augmente les frais opérationnels, le contrôle direct des ressources informatiques, des configurations, des comportements de mise à l'échelle et de l'utilisation des ressources peut ouvrir de nouvelles opportunités d'optimisation et de réduction des coûts. Par exemple, il peut être judicieux pour les équipes de commencer leur parcours de test de charge avec une architecture AWS Fargate sans serveur, puis de passer ultérieurement à l'autogestion des nœuds sous-jacents dans un cluster Amazon EKS.

Étapes d’implémentation

- Mettre à jour les scénarios de test : créez et mettez à jour en permanence des scénarios de test de charge pour valider les nouvelles fonctionnalités et les fonctionnalités refactorisées, et vérifier qu'elles reflètent l'état actuel du jeu.
- Évaluez les frameworks de test de charge : adaptez-vous aux nouveaux frameworks selon les besoins pour simuler la charge utilisateur, prendre en charge de nouveaux protocoles et vous aligner sur l'expertise et les chaînes d'outils de l'équipe.
- Optimisez les coûts : commencez par les AWS services gérés pour des raisons de facilité et de commodité, puis envisagez une infrastructure autogérée pour réaliser des économies au fur et à mesure que l'équipe se familiarisera avec le processus de test de charge.

Ressources

Consultez les ressources suivantes pour en savoir plus sur nos meilleures pratiques liées à l'excellence opérationnelle.

Documentation et blogs

- Bonnes pratiques en matière d'architecture pour les technologies du jeu
- Gérer votre studio de jeu sur le AWS pt.1
- Gérer votre studio de jeu sur AWS pt. 2
- Gérer votre studio de jeu sur AWS pt. 3
- Mise en place de votre AWS environnement de bonnes pratiques
- stratégie multi-comptes pour votre zone de landing zone de Control Tower
- Pipeline d'analyse des jeux
- Optimisez vos informations sur les données de jeu grâce à Game Analytics Pipeline
- Tirer parti AWS Glue d'Amazon Redshift Spectrum pour obtenir des informations sur les joueurs
- Comment configurer un CI/CD pipeline sur
- Comment Good Job Games accélère de 43 % grâce à AWS Build Pipeline
- Implémentation d'un pipeline de génération pour les applications mobiles Unity
- Autres CI/CD blogs pertinents
- Jeu DevOps simplifié avec le blog Game-Server CD Pipeline

- Harmony Games déploie un backend de jeu entièrement personnalisé en utilisant AWS Cloud Development Kit (AWS CDK) ()AWS CDK
- GameLiftPréparer le lancement
- Hébergement de serveurs de jeux hybrides avec Amazon Gamelift Anywhere
- Accélérez le développement de serveurs de jeux avec Amazon Gamelift Anywhere et Amazon Gamelift Agent
- Comment héberger votre jeu Unreal Engine pour moins de 1$ par joueur avec Amazon Gamelift
- Nouveau guide de solution pour créer des backends de jeu multiplateformes évolutifs sur AWS
- Chargement du moteur de jeu principal Pragma auprès d'un million d'utilisateurs simultanés sur AWS
- Comment Code Wizards a testé Nakama d'Heroic Lab auprès de deux millions de joueurs simultanés avec AWS
- Bonnes pratiques en matière d'unités organisationnelles
- AWS X-Ray
- AWS Compte à rebours
- AWS pour Games Solutions Hub

Solutions partenaires

- New Relic
- Splunk APM
- BackTrace.io
- Sentinelle
- APM Datadog
- Honeycomb.io

Livres blancs

- Organisation de votre environnement à l'aide de plusieurs comptes
- Présentation des modèles de développement de jeux évolutifs sur AWS

Vidéos

- YouTubesérie : Building Games on AWS
- AWS pour les jeux : Boss LEVEL Podcast
- Re:Invent 2023 : mise à l'échelle AWS des 10 premiers millions d'utilisateurs
- Re:Invent 2022 : comment Riot Games traite 20 To d'analyses par jour sur AWS
- Re:Invent 2022 : How AWS et Riot Games ont créé un moteur de reporting sur la gouvernance
- Re:Invent 2023 : Implémentation de modèles de conception distribués sur AWS
- Re:Invent 2023 : transformer un jeu multijoueur en millions avec Mortal Kombat 1
- Re:Invent 2022 : L'évolution de l'ingénierie du chaos chez Netflix
- Re:Invent 2023 : meilleures pratiques pour la gouvernance du cloud
- Re:Invent 2023 : Bonnes pratiques pour créer des architectures multirégionales sur AWS

Supports de formation

- Curriculum — Premiers pas avec AWS For Games, partie 1

Sécurité

Le pilier de la sécurité inclut la capacité de protéger les informations, les systèmes et les actifs tout en apportant de la valeur commerciale grâce à l'évaluation et à l'atténuation des risques. En raison de leur visibilité mondiale et du grand nombre de joueurs, les jeux constituent une cible privilégiée pour les exploiteurs, les pirates informatiques et les autres personnes qui cherchent des moyens d'exploiter et d'abuser des systèmes. Cela peut souvent se traduire par une expérience de jeu décevante et une augmentation des coûts pour le développeur du jeu si aucune base de sécurité solide n'est mise en place.

Comme décrit dans le modèle de responsabilité partagée, il est important de comprendre quels aspects de la sécurité relèvent de la responsabilité du client AWS et quels aspects relèvent de la responsabilité du client afin d'être prêt à maintenir une solide posture de sécurité. Ce pilier fournit des conseils sur les meilleures pratiques en matière de sécurité dans le cloud que vous devez prendre en compte lors du développement et de l'exploitation de jeux dans le cloud.

Avant de concevoir un système, vous devez établir un ensemble de bonnes pratiques de sécurité qui incluent des contrôles d'accès. En outre, vous devriez être en mesure d'identifier les incidents de sécurité et de protéger vos systèmes et services tout en préservant la confidentialité et l'intégrité des données grâce à la protection des données. Vous devez disposer d’un processus bien défini et utilisé pour répondre aux incidents de sécurité. Ces outils et techniques sont importants car ils soutiennent les objectifs commerciaux tels que la prévention des pertes financières ou le respect des obligations réglementaires.

Exemple client

AnyCompany Games est un studio de jeu fictif qui est en train d'améliorer sa sécurité. La sécurité peut être facile à comprendre lorsqu'il existe une explication de son application directe. AnyCompany Les jeux sont utilisés dans cette section pour contextualiser les meilleures pratiques de sécurité décrites dans le pilier.

Domaines prioritaires

- Principes de conception
- Bases de la sécurité
- Sécurité permanente
- Gestion des identités et des accès

Principes de conception

Outre les principes de conception énoncés dans le pilier sécurité du livre blanc Well-Architected Framework, les principes de conception suivants peuvent renforcer la sécurité de votre charge de travail de jeu dans le cloud :

- Surveillez et modérez le comportement d'utilisation des joueurs : capturez et analysez les données d'utilisation pour comprendre comment les joueurs interagissent avec votre jeu et les fonctionnalités sociales. En analysant ces données, vous pouvez détecter et réagir aux comportements abusifs et inappropriés susceptibles de dégrader l'expérience des joueurs.

Bases de la sécurité

GAMESEC01 : Comment implémentez-vous les principes fondamentaux de sécurité pour le développement de jeux ?

Les studios de jeux vidéo ont besoin d'une approche de sécurité unique qui protège à la fois les environnements de développement et les services aux joueurs en direct. Une stratégie AWS de sécurité robuste pour les studios de jeux nécessite trois éléments interconnectés : une structure multi-comptes, une authentification forte et une stratégie d'autorisation claire utilisant des politiques IAM. Une AWS structure multi-comptes permet aux studios de séparer les différents projets de jeu, étapes de développement et environnements d'outils. Cela permet aux studios de contrôler de manière plus précise des éléments tels que l'accès à des environnements ou à des services spécifiques. L'activation de l'authentification renforcée garantit que les membres de l'équipe peuvent

accéder en toute sécurité aux ressources de développement, qu'ils travaillent en studio ou à distance, tout en maintenant des contrôles stricts sur le code source, les builds de jeux et les outils propriétaires. Les studios devraient également avoir une stratégie d'autorisation claire pour accorder des autorisations en utilisant le principe du moindre privilège avec les autorisations et les rôles IAM. Utilisez les rôles IAM pour attribuer des autorisations entre les différents rôles de l'équipe de développement, par exemple en donnant aux équipes de développement l'accès à des AWS services de bas niveau tout en limitant les artistes et les concepteurs à des systèmes de gestion des actifs et de création spécifiques. Cette approche spécialisée permet aux studios de jeux de protéger leur propriété intellectuelle, de maintenir des flux de travail de développement efficaces et de faire évoluer leurs équipes en toute sécurité, tout en donnant aux développeurs l'accès approprié pour itérer rapidement sur leurs projets.

Bonnes pratiques

- GAMESEC01-BP01 Utilisez des rôles et un accès fédéré, plutôt que l'utilisateur root du compte, pour effectuer des actions sur votre environnement AWS
- GAMESEC01-BP02 AWS Control Tower À utiliser pour configurer rapidement un environnement multi-comptes sur AWS
- GAMESEC01-BP03 Utiliser des politiques de rôle au moindre privilège adaptées à des fonctions professionnelles spécifiques
- GAMESEC01-BP04 Utilisez des rôles et des politiques d'accès fédérées ainsi que des politiques d'accès au niveau du compte pour accorder l'accès à vos ressources AWS
- GAMESEC01-BP05 Utiliser un fournisseur d'identité central

GAMESEC01-BP01 Utilisez des rôles et un accès fédéré, plutôt que l'utilisateur root du compte, pour effectuer des actions sur votre environnement AWS

Lorsque vous créez un Compte AWS, vous commencez par une identité connue sous le nom d'utilisateur root, accessible à l'aide de l'adresse e-mail et du mot de passe associés au compte. L'utilisateur root dispose d'un accès complet aux AWS services et aux ressources de ce compte. Dans la plupart des cas, vous devez éviter d'utiliser l'utilisateur root pour les day-to-day tâches. Lorsqu'un accès au niveau du root est requis, confirmez qu'il est absolument nécessaire et vérifiez qu'une journalisation et des garde-fous supplémentaires sont en place pour suivre son utilisation.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Dans une AWS Organizations configuration, chaque compte possède toujours son propre utilisateur root, mais l' day-to-dayaccès doit plutôt être géré par le biais des rôles IAM et des utilisateurs de l'IAM Identity Center. Créez un accès basé sur les rôles adapté aux étapes du cycle de vie de votre jeu et aux équipes. Par exemple, l'équipe chargée des opérations en direct peut avoir besoin d'autorisations pour gérer les événements en jeu, tandis que les développeurs doivent avoir accès aux mises à jour push. Lorsque vous travaillez avec des services ou des partenaires tiers, utilisez l'accès fédéré pour permettre une collaboration sécurisée sans exposer l'infrastructure sensible. Cette approche permet de vérifier que chaque utilisateur ou partenaire ne dispose que de l'accès dont il a besoin, tout en préservant la sécurité de l'infrastructure de votre jeu et des données des joueurs.

Exemple client

AnyCompany Les jeux ont mis en place un contrôle d'accès basé sur les rôles lors du développement de leur nouveau jeu. En utilisant des rôles IAM spécifiques pour leurs diverses équipes de développement, ils évitent d'utiliser des informations d'identification partagées. Cette configuration permet à une équipe de développement d'assumer un rôle dans les principaux systèmes de jeu, tandis que le rôle de l'équipe chargée du contenu est uniquement d'accéder aux services de gestion des actifs.

Étapes d’implémentation

- N'utilisez pas l'utilisateur root après avoir créé un compte, sauf si cela est absolument nécessaire. Créez le compte, sécurisez l'utilisateur root, créez immédiatement les rôles d'administration IAM requis et attribuez ce rôle à un utilisateur fédéré.
- N'utilisez l'utilisateur root que lorsque vous devez effectuer un nombre limité de tâches qui ne sont accessibles qu'à l'utilisateur root. Ces tâches incluent par exemple la modification de l'adresse e-mail de votre utilisateur root et la modification de votre plan de AWS support.

GAMESEC01-BP02 AWS Control Tower À utiliser pour configurer rapidement un environnement multi-comptes sur AWS

Si vous commencez à l'utiliser AWS avec un seul compte, vous constaterez peut-être que votre studio de jeu s'en servira au fur et à mesure que votre processus de développement de jeux avance. Par exemple, avec un seul Compte AWS, vous pourriez commencer à atteindre les limites de service, ou vos coûts pour différents projets et charges de travail peuvent devenir plus complexes.

La création de différents comptes pour différents titres de jeu et environnements permet aux équipes d'expérimenter de nouvelles fonctionnalités, de contourner les limites de service et de maintenir le niveau de sécurité et la conformité. En mettant en œuvre une stratégie multi-comptes dans AWS, vous pouvez bénéficier de la répartition des limites de service sur plusieurs comptes et avoir un aperçu de vos AWS coûts.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

On pense souvent à tort que l'utilisation de plusieurs Comptes AWS sera automatiquement plus confuse et prendra plus de temps. Au contraire, l'utilisation de AWS services conçus pour faciliter la gouvernance de plusieurs comptes peut aider votre studio de jeu à passer moins de temps à gérer vos comptes.

Vous pouvez utiliser AWS Control Tower ce service pour approvisionner en toute sécurité un AWS environnement multi-comptes. Control Tower est recommandé si vous créez un nouvel AWS environnement, si vous commencez votre voyage AWS ou si vous êtes totalement novice AWS. Au cours du court processus de configuration, vous pouvez intégrer d'autres AWS services impliqués dans la gestion des comptes et de l'accès des utilisateurs AWS Organizations, tels que Service Catalog et AWS IAM Identity Center.

Exemple client

AnyCompany Les jeux fonctionnaient initialement à partir d'un seul jeu Compte AWS, mais ils se sont heurtés à de nombreux obstacles lorsque l'un des membres de l'équipe de développement de leurs jeux a atteint les limites de EC2 service lors d'un bêta-test crucial. Dans le même temps, l'équipe de développement d'un autre jeu a eu du mal à allouer des ressources pour son pipeline de tests automatisés. La situation a atteint un point de rupture lorsque AnyCompany Games n'a pas pu correctement séparer les coûts entre les projets, ce qui a rendu difficile le budgétisation du développement de chaque jeu.

AnyCompany Games a ensuite mis en œuvre une stratégie multi-comptes en utilisant AWS Control Tower. Ils ont créé des comptes distincts pour chaque projet de jeu, avec des environnements de développement, d'assurance qualité et de production distincts. Cette séparation au niveau du compte isole les données et les actifs de chaque projet, de sorte que les équipes travaillant sur un jeu ne peuvent pas accéder aux ressources d'un autre jeu ou les modifier. Ils ont ainsi établi une structure de facturation centralisée indiquant clairement les coûts d'infrastructure de chaque jeu et ont également créé des politiques d'accès à l'échelle de l'organisation. AWS Organizations

Étapes d’implémentation

- Utilisez la tour de AWS contrôle pour configurer un environnement multi-comptes automatisé.
- Organisez les comptes en fonction des environnements (tels que le développement, l'assurance qualité et la production).
- Utilisez AWS IAM Identity Center et Service Catalog pour centraliser les autorisations des utilisateurs et rationaliser le provisionnement des ressources entre les comptes.

GAMESEC01-BP03 Utiliser des politiques de rôle au moindre privilège adaptées à des fonctions professionnelles spécifiques

La configuration des politiques IAM est essentielle pour établir une base de sécurité solide. Lorsque vous définissez des autorisations avec des stratégies IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Par exemple, les équipes d'assurance qualité ont besoin d'un accès pour changer les choses dans les environnements de test, mais ne doivent pas être en mesure de modifier l'environnement de production.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Vous pouvez commencer par des autorisations générales, telles que les politiques gérées, tout en explorant les autorisations requises pour votre charge de travail ou votre cas d'utilisation. Au fur et à mesure que votre cas d’utilisation évolue, vous pouvez réduire les autorisations que vous accordez pour obtenir le moindre privilège.

Étapes d’implémentation

- Suivez la pratique des autorisations du moindre privilège pour créer des rôles IAM pour les utilisateurs et les applications.
- Utilisez des politiques AWS gérées pour fournir rapidement un accès étendu tout en identifiant les autorisations spécifiques dont les équipes ou les applications ont besoin pour effectuer leurs tâches.
- Les studios peuvent également utiliser la génération de politiques de l'analyseur d'accès IAM pour générer des politiques IAM personnalisées basées sur CloudTrail des événements identifiant les actions et les services utilisés par une entrée IAM.

- Passez régulièrement en revue les politiques IAM et modifiez les politiques trop permissives.

GAMESEC01-BP04 Utilisez des rôles et des politiques d'accès fédérées ainsi que des politiques d'accès au niveau du compte pour accorder l'accès à vos ressources AWS

AWS Les nouveaux utilisateurs utilisent souvent les politiques IAM uniquement lorsqu'ils accordent l'accès à d'autres personnes. Toutefois, si vous en utilisez AWS Organizations, réfléchissez à la manière d'utiliser les politiques de contrôle des services conjointement avec les politiques IAM pour accorder aux membres de l'équipe de votre studio et aux sous-traitants les niveaux d'accès nécessaires.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Vous pouvez créer des politiques IAM pour autoriser ou refuser l'accès aux AWS services ou aux actions d'API compatibles avec Gestion des identités et des accès AWS. Ils ne peuvent être appliqués qu'aux identités IAM, telles que les utilisateurs, les groupes ou les rôles. Par exemple, une politique IAM peut être utilisée pour fournir à un utilisateur un accès en lecture seule à Amazon S3.

Les politiques de contrôle des services (SCPs) sont des garde-fous pour vous. Comptes AWS Un SCP n'accorde pas d'autorisations, elles sont utilisées pour restreindre les actions sur les AWS services pour les comptes de membres individuels. Par exemple, un SCP peut refuser Compte AWS à un l'accès à une région particulière.

Lorsqu'une action est entreprise, la politique IAM correspondante est évaluée en combinaison avec le SCPs. Comme dans l'exemple précédent, si un rôle tente d'exécuter une EC2 instance, IAM indique s'il est autorisé (« Autoriser » pour ec2 :RunInstances) et SCPs déterminera si son choix de région est valide (« us-east-1 » est autorisé, mais « us-west-1 » est refusé par le SCP).

En superposant les politiques IAM, SCPs vous pouvez vérifier que toute personne accédant à vos AWS ressources ne recevra que les autorisations appropriées dont elle a besoin. Cela est particulièrement important à prendre en compte si vous Comptes AWS et vos ressources couvrent plusieurs régions, mais que tous les membres de votre studio de jeu n'ont pas besoin d'y accéder à toutes.

Vous pouvez adapter les politiques IAM pour accorder à des équipes spécifiques des autorisations spécifiques pour mettre à jour des éléments tels que les configurations de jeu, la gestion des

données des joueurs, la configuration d'événements promotionnels et la modération du contenu généré par les utilisateurs. En attendant, vous pouvez l'utiliser SCPs pour appliquer des contrôles à l'échelle de l'organisation essentiels au fonctionnement du jeu. Il peut s'agir notamment de restreindre le déploiement aux seules régions approuvées dans lesquelles le jeu fonctionne, d'empêcher l'accès non autorisé aux banques de données sensibles des joueurs, de faire respecter les exigences de conformité et de contrôler les coûts en limitant l'utilisation des services sur les comptes de développement.

Étapes d’implémentation

- Utilisez les politiques IAM pour gérer les autorisations pour des utilisateurs, des groupes ou des rôles individuels.
- Utilisez les politiques de contrôle des services (SCPs) AWS Organizations pour appliquer les autorisations au niveau du compte.
- Combinez les politiques IAM et SCPs accordez uniquement l'accès requis à des utilisateurs et à des comptes spécifiques.

Ressources

- Politiques et autorisations dans AWS IAM
- Politiques de contrôle des services
- Stratégies gérées par AWS pour les activités professionnelles

GAMESEC01-BP05 Utiliser un fournisseur d'identité central

Un fournisseur d'identité central agit comme une source unique pour le stockage et la gestion des informations d'identification, des identités, des autorisations et de l'authentification des utilisateurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Utilisez un fournisseur d'identité central pour rationaliser votre processus d'authentification des utilisateurs, appliquer des politiques de sécurité cohérentes et simplifier la gestion des utilisateurs dans l'ensemble de vos applications Comptes AWS et applications. Grâce à une approche centralisée, il n'est plus nécessaire de gérer séparément les identités et les informations d'identification des utilisateurs, ce qui réduit le risque d'incohérences, de redondances et d'autres

failles de sécurité. La consolidation des identités et de l'authentification des utilisateurs en un seul endroit permet également d'améliorer la visibilité, le contrôle et l'auditabilité de AWS l'ensemble de votre environnement.

Exemple client

AnyCompany Les jeux se sont heurtés à d'importants défis en matière de gestion de l'accès des développeurs sur l'ensemble de leur AWS infrastructure en pleine expansion. Leur équipe de développement est passée de 50 à 200 personnes réparties sur trois titres majeurs. Au départ, chaque équipe de projet gérait ses propres identifiants d' AWS accès, ce qui entraînait des pratiques de sécurité incohérentes, des retards dans l'intégration des nouveaux développeurs et des incidents de sécurité occasionnels.

Le studio a implémenté AWS IAM Identity Center en tant que fournisseur d'identité central, consolidant ainsi la gestion des utilisateurs dans un système unique. Ils l'ont connecté à leur annuaire d'entreprise existant, permettant aux développeurs d'utiliser les mêmes informations d'identification d'entreprise pour AWS y accéder. Les développeurs utilisent désormais leur identifiant unique d'entreprise existant pour obtenir l' AWS accès dont ils ont besoin pour terminer leur travail

Étapes d’implémentation

- Envisagez d'utiliser AWS IAM Identity Center comme fournisseur d'identité central. Cela permet une gestion cohérente des accès dans l'ensemble de votre entreprise Comptes AWS, fournit à vos employés une authentification unique et simplifie l'audit de l'accès des utilisateurs à vos AWS applications. IAM Identity Center se connecte également aux identités d'entreprise existantes des fournisseurs d'identité pris en charge.

Sécurité permanente

GAMESEC02 : Comment appliquez-vous, maintenez-vous et surveillez-vous les meilleures pratiques de sécurité continues ?

Le respect des meilleures pratiques de sécurité est essentiel pour les entreprises de tous les secteurs, mais en particulier dans le secteur du jeu vidéo. L'industrie du jeu vidéo repose sur le maintien et le maintien de la confiance des joueurs et sur la création d'une solide réputation, et même des problèmes de sécurité mineurs peuvent rapidement saper cette confiance.

En outre, la nature mondiale de l'industrie du jeu nécessite le respect des diverses réglementations et normes du secteur régissant la protection des données, la confidentialité des consommateurs et la sécurité dans les régions où les jeux sont proposés. Un gameplay équitable et sécurisé est un autre aspect essentiel qui souligne l'importance de mesures de sécurité robustes. La triche, le piratage et les autres formes d'exploitation du jeu peuvent perturber l'expérience de jeu des joueurs légitimes. Des contrôles de sécurité stricts sont donc essentiels pour préserver l'intégrité du jeu et favoriser des règles du jeu équitables pour les participants.

Bonnes pratiques

- GAMESEC02-BP01 Utiliser des modèles prêts à déployer pour les pratiques de sécurité standard
- GAMESEC02-BP02 Utiliser des techniques de correction automatisées lorsqu'un événement de sécurité survient

GAMESEC02-BP01 Utiliser des modèles prêts à déployer pour les pratiques de sécurité standard

Ready-to-deployles modèles constituent un moyen proactif et agile d'évaluer votre niveau de sécurité dans le cloud. Les modèles préconfigurés évaluent la sécurité de votre cloud et mettent en œuvre rapidement les modifications nécessaires. Les modèles incluent un large éventail de bonnes pratiques issues de diverses technologies et de cadres de sécurité largement acceptés. L'utilisation de modèles peut aider les studios de jeux à maintenir des configurations d'infrastructure cohérentes, d'autant plus qu'ils peuvent évoluer et en ajouter d'autres Comptes AWS pour prendre en charge de nouvelles charges de travail.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

En utilisant AWS des services et en implémentant des ready-to-deploy modèles, les développeurs de jeux peuvent évaluer et renforcer de manière proactive leur niveau de sécurité dans le cloud, en préservant leur propriété intellectuelle, en protégeant les données des joueurs et en favorisant un environnement de jeu sécurisé grâce à des évaluations de sécurité régulières et à une surveillance continue pour identifier et corriger rapidement les vulnérabilités potentielles.

Exemple client

AnyCompany Les jeux ont été confrontés à un défi de taille lors de la préparation du lancement de leur prochain titre dans l'industrie européenne. Ils ont réalisé que leurs pratiques de traitement des

données existantes ne répondaient pas aux exigences du RGPD. Ils se sont tournés vers AWS Security Hub CSPM AWS Config et ses ready-to-deploy modèles pour trouver une solution. L'équipe a mis en œuvre le pack de conformité spécifique au RGPD en AWS Config, qui a automatiquement évalué son infrastructure existante par rapport aux normes du RGPD. Cette analyse initiale a révélé plusieurs lacunes critiques, telles que des politiques de conservation des données inappropriées et des contrôles d'accès inadéquats sur l'endroit où les données des joueurs étaient stockées. En utilisant les règles prédéfinies du modèle, AnyCompany Games a rapidement mis en œuvre les modifications nécessaires. De plus, les contrôles de conformité automatisés continus fournis par le modèle ont permis à la petite équipe de maintenir la conformité au RGPD sans effort, tout en continuant à mettre à jour et à développer le jeu.

Étapes d’implémentation

- Utilisez des modèles pour les pratiques de sécurité standard, telles que les règles gérées et les packs de conformité intégrés AWS Config et les normes dans AWS Security Hub CSPM.
- Passez en revue les détails des normes CSPM du Security Hub afin de déterminer celles qui correspondent le mieux aux besoins de sécurité de votre studio de jeu.

GAMESEC02-BP02 Utiliser des techniques de correction automatisées lorsqu'un événement de sécurité survient

À l'aide de techniques de correction automatisées, les développeurs de jeux peuvent protéger et entretenir de manière proactive leur infrastructure de jeu et minimiser l'impact potentiel d'un incident de sécurité. Si un problème de sécurité est détecté, utilisez un runbook pour vous aider à réagir à la situation. Automatisez ces réponses dans la mesure du possible afin de résoudre les problèmes plus rapidement et de réduire leur impact. Cela améliore l'expérience des joueurs en réduisant les risques de temps d'arrêt et de perturbations du jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Le fait de se préparer à répondre aux problèmes de sécurité permet non seulement de préserver l'expérience des joueurs, mais également de respecter les différentes normes réglementaires et de conformité. En outre, l'utilisation de réponses de sécurité automatisées permet d'adapter vos opérations de sécurité à mesure que vos charges de travail augmentent. AWS fournit des services permettant d'identifier et d'automatiser une réponse à ces incidents.

Exemple client

AnyCompany Les jeux ont été confrontés à un incident de sécurité critique lorsqu'un bucket S3 contenant des modèles de personnages et des textures inédits pour leur prochain jeu a été accidentellement rendu public lors d'une mise à jour de routine du pipeline d'actifs. Le système de sécurité automatique a détecté le changement d'autorisation du bucket quelques minutes après la modification. Le système a immédiatement exécuté son programme de correction : retour du bucket au statut privé, enregistrement des tentatives d'accès pendant la période d'exposition, notification à l'équipe de sécurité et création d'un CloudTrail journal détaillé des modifications d'autorisation.

Étapes d'implémentation

- Utilisez la AWS solution Automated Security Response on pour implémenter des runbooks d'automatisation qui définissent les actions qui seront automatiquement entreprises en réponse aux événements de sécurité survenus dans AWS Security Hub CSPM.

Ressources

- AWS pour Games Blog — Gérer votre studio de jeu sur AWS : Partie 1
- AWS pour Games Blog — Gérer votre studio de jeu, AWS partie 2
- Enregistrez une unité organisationnelle existante auprès de AWS Control Tower
- Compte AWS utilisateur root
- Tâches nécessitant les informations d'identification de l'utilisateur root
- Réponse de sécurité automatisée sur AWS

Gestion des identités et des accès

GAMESEC03 : Comment gérez-vous l'identité des joueurs et la gestion des accès ?

Lorsque vous développez un jeu, vous devez déterminer comment vous allez fournir aux joueurs l'accès à votre jeu et aux services associés. La section suivante explore les considérations de conception, notamment l'authentification des joueurs, l'autorisation et l'authentification multifactorielle.

Bonnes pratiques

- GAMESEC03-BP01 Déterminez votre approche pour identifier et contrôler l'accès des joueurs à l'environnement et aux ressources de votre jeu
- GAMESEC03-BP02 Authentifiez les demandes envoyées au service de backend de votre jeu
- GAMESEC03-BP03 Utilisez le service de backend de votre jeu pour valider les demandes des joueurs souhaitant rejoindre une partie multijoueur
- GAMESEC03-BP04 Appliquer une politique de sécurité stricte pour les comptes utilisateurs des joueurs en exigeant un mot de passe fort
- GAMESEC03-BP05 Offrir aux joueurs la possibilité de configurer l'authentification multifactorielle (MFA) sur leurs comptes

GAMESEC03-BP01 Déterminez votre approche pour identifier et contrôler l'accès des joueurs à l'environnement et aux ressources de votre jeu

Cette décision est influencée par votre stratégie d'acquisition et de monétisation de joueurs, l'expérience des joueurs et d'autres facteurs tels que les fonctionnalités existantes qui peuvent être fournies par vos partenaires éditeurs de jeux. Par exemple, un jeu peut nécessiter des achats et obliger un joueur à créer un profil utilisateur pour associer des méthodes de paiement en argent réel à son compte.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Un jeu peut également souhaiter réduire les obstacles à l'entrée pour les joueurs qui jouent pour la première fois en supprimant la nécessité de créer un compte utilisateur avant de jouer au jeu, augmentant ainsi les chances qu'un joueur essaie le jeu pour la première fois. Généralement, les jeux mettent en œuvre une ou plusieurs combinaisons d'approches de gestion de l'identité et de l'accès des joueurs pour leur jeu.

Accès anonyme ou non authentifié

Ce niveau d'accès est utile dans les situations où un jeu n'oblige pas le joueur à créer un nouveau compte utilisateur ou à établir un lien avec son identité sur les réseaux sociaux et les systèmes de jeu. C'est le moyen le plus simple et le plus rapide pour un joueur de commencer à jouer. Il est particulièrement utile dans les jeux mobiles où un développeur de jeux peut souhaiter réduire les obstacles à l'entrée lors de l'expérience initiale.

Dans ce scénario d'accès, si vous souhaitez identifier l'utilisation depuis l'installation du jeu, vous pouvez programmer le client du jeu pour qu'il génère et stocke un identifiant unique sur l'appareil du joueur. Cet identifiant unique est utilisé pour identifier le joueur au fil des sessions de jeu sur son appareil et permettre d'établir des rapports d'analyse sur l'utilisation au fil du temps. Plus tard, si un joueur choisit de créer un compte, vous pourrez associer son nouveau compte utilisateur à son identifiant unique généré précédemment. Cela associera leur nouvelle identité de joueur à leur historique d'utilisation, qui peut inclure des statistiques et des succès dans le jeu.

Si un joueur ne crée pas et n'associe pas de compte, l'appareil qu'il utilise pour interagir avec le jeu peut être identifié de manière unique, mais les informations récupérables sur le joueur ne sont ni collectées ni stockées. Ainsi, si le joueur casse ou perd son appareil, les données précédemment stockées associées à l'appareil sont également perdues et risquent de ne pas être récupérables.

Authentification avec nom d'utilisateur et mot de passe

Un jeu peut permettre aux joueurs de créer leur propre compte utilisateur avec un nom d'utilisateur et un mot de passe qui sont stockés dans le backend du jeu. Cela peut se produire lorsqu'un développeur de jeux collabore avec un éditeur de jeux qui possède déjà un système de compte joueur existant auquel le développeur peut s'intégrer. Un développeur qui publie ses propres jeux peut également souhaiter simplifier l'expérience des joueurs en permettant aux joueurs de créer un compte utilisateur unique pour accéder à tous les jeux qu'ils publient.

Authentification et association de comptes à des réseaux sociaux et à des systèmes de jeu tiers

Il est courant que les jeux en ligne et les jeux dotés de fonctionnalités sociales proposent une fédération de fournisseurs d'identité tiers afin de simplifier l'expérience des joueurs. Au lieu de demander aux joueurs de créer une combinaison de nom d'utilisateur et de mot de passe pour s'authentifier, vous pouvez utiliser la fédération d'identité pour permettre aux joueurs de s'authentifier à l'aide de leurs comptes tiers sur les réseaux sociaux et les systèmes de jeu. Ce processus de connexion simplifie l'expérience de connexion et d'inscription pour les joueurs. Il fournit également une alternative pratique à la création obligatoire de compte et une méthode fluide pour les joueurs pour accéder aux jeux.

Pour les développeurs de jeux, un processus de connexion fédéré peut simplifier le flux de travail de vérification des joueurs. Cela peut également fournir un moyen plus fiable de gérer les données des joueurs utilisées à des fins de personnalisation. En effet, vous n'avez pas besoin de demander aux joueurs de vous fournir certaines données qu'ils ont probablement déjà fournies au fournisseur d'identité tiers. De plus, ces systèmes intègrent des fonctionnalités sociales supplémentaires, telles que la possibilité de relier les joueurs à leurs amis.

Étapes d’implémentation

- Utilisez un accès anonyme ou non authentifié pour réduire les obstacles auxquels sont confrontés les nouveaux joueurs en générant un identifiant d'appareil unique pour suivre l'utilisation et en permettant de lier les comptes ultérieurement.
- Mettez en œuvre l'authentification par nom d'utilisateur et mot de passe pour les comptes utilisateurs dédiés, en utilisant les systèmes de comptes de joueurs existants ou en créant une expérience unifiée dans tous les jeux.
- Intégrez des fournisseurs d'identité tiers pour une authentification fédérée, simplifiant les processus de connexion et permettant l'accès aux fonctionnalités sociales et aux données de personnalisation.

GAMESEC03-BP02 Authentifiez les demandes envoyées au service de backend de votre jeu

L'authentification des demandes envoyées à votre service de backend de jeu peut empêcher les demandes indésirables de réussir.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Vous devez fournir un service d'authentification permettant aux joueurs de se connecter, qui doit renvoyer des jetons sécurisés de courte durée, tels qu'un jeton Web JSON (JWT), au client du jeu lorsqu'un joueur s'authentifie avec succès.

Ces jetons peuvent inclure des assertions de réclamation contenant les attributs des joueurs et d'autres métadonnées pertinentes. Ces métadonnées pertinentes peuvent être utilisées dans les demandes ultérieures envoyées par le client du jeu à votre backend de jeu pour authentifier les demandes et les autoriser dans le contexte du joueur authentifié.

Vous pouvez soit concevoir et créer votre propre système d'authentification des joueurs, ce qui nécessiterait une amélioration et une maintenance continues, soit utiliser les fonctionnalités évolutives et sécurisées d'inscription, de connexion et de contrôle d'accès des utilisateurs proposées par Amazon Cognito.

Les groupes d'utilisateurs Amazon Cognito incluent un répertoire d'utilisateurs pour l'authentification et l'autorisation. Un pool d'utilisateurs vous permet APIs d'intégrer à votre jeu des flux de travail

d'inscription, de connexion et de réinitialisation du mot de passe, qui peuvent être intégrés à des fournisseurs d'identité tiers. Les équilibreurs de charge d'application et Amazon API Gateway proposent tous deux des intégrations à Cognito afin d'intégrer l'authentification des utilisateurs pour les demandes envoyées à vos backends de jeu personnalisés hébergés par ces services.

Si votre jeu prend en charge l'accès anonyme et que vous ne pouvez pas authentifier un joueur, vous pouvez utiliser une approche d'authentification client pour garantir une expérience plus sécurisée lors de l'intégration à votre backend de jeu. Si votre client de jeu utilise directement des AWS services, les demandes adressées à ces services doivent être signées à l'aide d'informations d'identification. Pour fournir des informations d'identification à votre client de jeu aux utilisateurs non authentifiés, vous pouvez utiliser le AWS SDK pour récupérer des informations d'identification de courte durée provenant des pools d'identités Amazon Cognito, qui peuvent être utilisées pour signer vos demandes de services. AWS Ces informations d'identification peuvent être actualisées depuis votre client de jeu.

Outre l'intégration directe au AWS SDK depuis le client du jeu, vous pouvez également créer votre propre backend de jeu à l'aide d'un service tel qu'Amazon API Gateway, qui prend en charge les autorisations personnalisées. En concevant votre propre service de backend de jeu, vous pouvez obtenir un contrôle autoritaire sur les requêtes grâce à une logique personnalisée côté serveur.

Pour plus d'informations sur la création d'un service principal pour les jeux hébergés via Amazon GameLift, consultez Concevez votre service client de jeu.

Exemple client

AnyCompany Les jeux ont renforcé la sécurité de leur prochain titre en adoptant une approche d'authentification et d'autorisation gérée. Au lieu de maintenir un système de nom d'utilisateur et de mot de passe personnalisé, ils ont utilisé des groupes d'utilisateurs Amazon Cognito pour gérer l'inscription et la connexion des joueurs, et des groupes d'identités pour permettre un accès anonyme aux joueurs essayant le mode entraînement avant de créer un compte. Ils ont également mis en place une logique d'autorisation personnalisée dans le jeu afin de reconnaître les rôles d'administrateur définis dans Cognito, permettant ainsi à ces utilisateurs d'accéder à des fonctionnalités de gestion spéciales dans le jeu.

Étapes d’implémentation

- Utilisez les groupes d'utilisateurs Amazon Cognito pour gérer l'authentification à l'aide de jetons sécurisés, notamment en activant des fonctionnalités telles que l'inscription JWTs, la connexion et la réinitialisation des mots de passe.

- Récupérez des informations d'identification de courte durée dans les pools d'identités Amazon Cognito afin que les utilisateurs anonymes puissent interagir en toute sécurité avec AWS les services.
- Implémentez des backends de jeu personnalisés à l'aide d'Amazon API Gateway pour une logique d'authentification personnalisée côté serveur.

GAMESEC03-BP03 Utilisez le service de backend de votre jeu pour valider les demandes des joueurs souhaitant rejoindre une partie multijoueur

Généralement, dans les jeux multijoueurs, un joueur rejoint une session de jeu en sélectionnant une option directement dans la liste des sessions disponibles, ou il soumet une demande pour trouver un match. Selon cette dernière approche, il incombe au développeur du jeu de localiser une session de jeu éligible et de fournir les informations de connexion (généralement une adresse IP et un numéro de port) au client de jeu du joueur. L'implémentation peut varier en fonction du genre de jeu que vous développez, mais quoi qu'il en soit, une bonne pratique de sécurité consiste à effectuer une validation côté serveur de la demande d'un joueur de rejoindre un jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Par exemple, dans un jeu multijoueur basé sur des sessions, la demande d'un joueur de rejoindre une session de jeu doit être validée par le logiciel de votre serveur de jeu auprès de votre service de matchmaking principal avant d'autoriser sa connexion au serveur. Lorsqu'un joueur demande à rejoindre une session de jeu, le serveur de jeu doit vérifier la demande pour obtenir un identifiant unique, tel qu'un identifiant de session du joueur et un ticket généré par le serveur qui a été précédemment fourni au client du jeu par le service de matchmaking de votre backend de jeu.

Lors de l'établissement de la connexion au serveur de jeu, votre logiciel côté serveur peut utiliser ces informations pour vérifier auprès du service de matchmaking que la demande de connexion du joueur est valide et vérifier que le joueur ne rejoint pas une place précédemment réservée dans la session de jeu à un autre joueur.

Pour les jeux hébergés sur Amazon GameLift, consultez client/server Interactions entre les jeux et les GameLift serveurs Amazon pour un exemple de la manière dont ce type de validation côté serveur peut être mis en œuvre.

Exemple client

Lors du lancement initial AnyCompany de la version bêta de Games, ils ont découvert que les joueurs contournaient leur système de matchmaking en se connectant directement aux serveurs de jeu, ce qui entraînait de graves problèmes d'intégrité concurrentielle. Lorsque les joueurs les mieux classés ont découvert qu'ils pouvaient partager les adresses IP des serveurs avec leurs amis, ils ont commencé à contourner le système de matchmaking basé sur les compétences, ce qui a amené des joueurs expérimentés à rejoindre des parties novices et à créer une expérience frustrante pour les nouveaux joueurs. AnyCompany Les jeux ont répondu en mettant en œuvre un système de validation côté serveur qui a généré des tickets de session uniques pour chaque demande de matchmaking. Le système exigeait que le joueur IDs et le matchmaking demandent des tickets et vérifient les tentatives de connexion avec leur service de matchmaking principal.

Étapes d'implémentation

- Validez les demandes d'adhésion de joueurs côté serveur à l'aide d'identifiants uniques tels que la session du joueur IDs et les tickets générés par le serveur.
- Confirmez la validité des demandes de connexion auprès du service de matchmaking pour bloquer les accès non autorisés.
- Vérifiez que les places réservées dans les sessions de jeu ne sont pas accessibles aux joueurs non autorisés pendant le processus de validation.

GAMESEC03-BP04 Appliquer une politique de sécurité stricte pour les comptes utilisateurs des joueurs en exigeant un mot de passe fort

Si un jeu offre aux joueurs la possibilité de créer un compte utilisateur avec un mot de passe, vous devez exiger que les mots de passe des joueurs respectent des politiques strictes. Par exemple, les groupes d'utilisateurs Amazon Cognito vous permettent de définir des exigences en matière de mot de passe pour les comptes utilisateurs. L'établissement d'une politique de mot de passe robuste peut protéger les comptes de vos joueurs contre le piratage par l'ingénierie sociale et les attaques par force brute.

Niveau d'exposition au risque si cette bonne pratique n'est pas respectée : élevé

Directives d'implémentation

Exemple client

AnyCompany Les jeux ont connu une crise lorsque leur célèbre titre a connu une vague de piratages de comptes en raison de politiques de mots de passe faibles. Les joueurs qui utilisaient des mots de

passe simples tels que « password123 » étaient victimes d'attaques automatisées par force brute, entraînant la perte d'objets et la compromission de la monnaie du jeu. Pour y remédier, AnyCompany Games a réorganisé son système de connexion et a exigé que les mots de passe ne soient jamais utilisés auparavant, qu'ils incluent au moins une lettre majuscule, un chiffre, un caractère spécial et une longueur minimale de 15 caractères.

Étapes d'implémentation

- Exigez des politiques de mot de passe strictes pour les comptes des joueurs afin d'améliorer la sécurité.
- Utilisez les groupes d'utilisateurs Amazon Cognito pour définir et appliquer les exigences relatives aux mots de passe.

GAMESEC03-BP05 Offrir aux joueurs la possibilité de configurer l'authentification multifactorielle (MFA) sur leurs comptes

Les comptes de joueurs peuvent être un atout pour les acteurs malveillants, en particulier dans les jeux qui proposent des devises et des achats intégrés au jeu. En raison de l'omniprésence du piratage des comptes des joueurs et des attaques d'ingénierie sociale, offrez aux joueurs la possibilité d'améliorer la sécurité de leurs comptes en configurant l'authentification multifactorielle (MFA).

Niveau d'exposition au risque si cette bonne pratique n'est pas respectée : moyen

Directives d'implémentation

Lorsqu'un joueur tente d'accéder à son compte en utilisant le MFA, un code temporaire est envoyé à son adresse e-mail, à son numéro de téléphone ou à une application mobile d'authentification multifactorielle spécialement conçue à cet effet. Pour s'authentifier avec succès, le joueur doit ensuite saisir le code dans le système de connexion dans un délai limité.

La MFA peut également être utilisée pour protéger les comptes qui tentent de s'authentifier à partir d'une nouvelle géolocalisation, les comptes signalés par l'assistance aux joueurs pour détecter d'éventuelles activités malveillantes, et même les comptes qui ne se sont pas connectés au jeu depuis longtemps.

Par exemple, les groupes d'utilisateurs Amazon Cognito peuvent configurer l'authentification multifactorielle sur les annuaires des utilisateurs.

Étapes d’implémentation

- Activez l'authentification multifactorielle (MFA) pour améliorer la sécurité des comptes des joueurs.
- Utilisez des codes temporaires envoyés par e-mail, téléphone ou via des applications MFA pour vérifier l'accès au compte.
- Appliquez le MFA pour les nouvelles géolocalisations, les comptes marqués ou les comptes présentant une inactivité prolongée.

Contrôle d’accès

GAMESEC04 : Comment bloquez-vous l'accès non autorisé au contenu du jeu ?

Les jeux modernes incluent une quantité importante de contenu, tel que le contenu téléchargeable (DLC), qui est un aspect important de l'engagement des joueurs et de la monétisation des jeux. Les joueurs s'attendent à un flot continu de nouveaux personnages, niveaux et défis, ce qui oblige les développeurs de jeux à répondre à la demande constante de nouveaux contenus pour fidéliser les joueurs. La variété et la taille du contenu peuvent varier considérablement en fonction du type de jeu et de l'appareil sur lequel le jeu est joué. Quel que soit le système du jeu, protégez le contenu du jeu contre tout accès non autorisé.

Bonnes pratiques

- GAMESEC04-BP01 Restreindre l'accès au contenu téléchargeable aux clients et utilisateurs autorisés
- GAMESEC04-BP02 Limiter l'accès d'origine aux réseaux de diffusion de contenu autorisés () CDNs
- GAMESEC04-BP03 Implémenter des restrictions géographiques pour limiter les accès non autorisés
- GAMESEC04-BP04 Restreignez l'accès au contenu avec des solutions de gestion des droits numériques (DRM)

GAMESEC04-BP01 Restreindre l'accès au contenu téléchargeable aux clients et utilisateurs autorisés

Limitez l'accès au contenu du jeu aux applications et aux clients autorisés. Envisagez d'utiliser Amazon S3 comme source rentable et évolutive pour stocker du contenu de jeu téléchargeable et Amazon CloudFront pour fournir du contenu performant aux joueurs dans le monde entier. Les deux services fournissent des mécanismes intégrés pour restreindre l'accès aux données stockées, par exemple pour restreindre l'accès aux utilisateurs authentifiés.

Niveau d'exposition au risque si cette bonne pratique n'est pas respectée : élevé

Directives d'implémentation

Octroi de l'accès au contenu stocké dans Amazon S3

Lorsque vous devez accorder l'accès à du contenu stocké dans S3, plusieurs facteurs doivent être pris en compte. Par défaut, seul celui Compte AWS qui a créé un compartiment S3 peut accéder aux objets qui y sont stockés. Pour accorder l'accès à vos applications internes et gérer le contenu stocké dans les compartiments Amazon S3, utilisez Gestion des identités et des accès AWS (IAM) pour créer des politiques fournissant un accès approprié.

Les rôles IAM peuvent être associés à des utilisateurs fédérés, à des systèmes ou à des applications hébergés dans des services, tels qu'Amazon EC2 AWS Lambda, et à des applications basées sur des conteneurs hébergées dans Amazon EKS et Amazon ECS. Par exemple, vous pouvez utiliser le AWS SDK ou AWS CLI pour publier et gérer des ressources de contenu de jeu dans des compartiments S3. Pour prendre en charge ce cas d'utilisation, vous pouvez créer un rôle IAM doté d'un accès approprié pour lire et écrire du contenu de jeu dans vos compartiments S3 et l'associer aux EC2 instances qui hébergent vos logiciels et scripts.

Des politiques basées sur les ressources peuvent être définies pour votre compartiment et pour des objets spécifiques. Les politiques de compartiment S3 sont associées à un compartiment S3 et peuvent être utilisées pour restreindre l'accès au compartiment et aux objets qu'il contient, ainsi que pour accorder l'accès à vos ressources Amazon S3 depuis d'autres comptes. Par exemple, dans les scénarios où plusieurs équipes ou studios de développement de jeux différents travaillent sur le même contenu de jeu et nécessitent le même accès au contenu hébergé de manière centralisée dans Amazon S3, vous pouvez utiliser une politique de compartiment S3 pour définir les autorisations d'accès entre comptes aux ressources S3. Envisagez d'utiliser des points d'accès S3, qui peuvent simplifier la gestion de l'accès aux données partagées en créant des points d'accès dotés de noms et d'autorisations spécifiques à chaque application ou ensemble d'applications. La documentation

Amazon S3 contient des bonnes pratiques supplémentaires pour le contrôle d'accès dans Amazon S3.

Granting short-term access to your content

Lorsque l'accès n'est nécessaire que pour une durée limitée, générez un accès temporaire URLs qui accorde un accès à court terme à votre contenu. Amazon S3 prend en charge la génération de fichiers présignés URLs, ce qui permet aux propriétaires d'objets d'accorder un accès limité dans le temps aux objets dans Amazon S3 sans mettre à jour votre politique de compartiment. Ainsi, l'utilisateur final ou l'application auxquels l'accès est accordé n'est pas obligé de disposer d'un compte ou d'autorisations IAM et utilise plutôt l'URL présignée pour accéder au contenu.

Il s'agit d'une bonne pratique couramment utilisée dans de nombreux cas d'utilisation des jeux, par exemple en accordant aux joueurs autorisés l'accès au contenu téléchargeable auquel ils ont droit et en fournissant un accès temporaire à du contenu de jeu d'une durée limitée. Presigned URLs peut également être utilisé pour fournir des autorisations temporaires pour le téléchargement de contenu dans un compartiment S3. Par exemple, vous pouvez envisager d'utiliser une URL présignée pour permettre à un joueur de télécharger les journaux des clients afin d'aider votre équipe d'assistance à résoudre un problème d'assistance concernant un joueur.

Utiliser un réseau de diffusion de contenu pour donner accès à votre contenu

Bien que vos applications, développeurs de jeux, artistes et autres membres du personnel puissent avoir besoin d'un accès direct au contenu des compartiments S3 à des fins de développement et de gestion, utilisez un réseau de diffusion de contenu pour fournir un accès au contenu accessible au public aux joueurs ou à d'autres utilisateurs sur Internet. Cette approche améliore les performances de téléchargement et réduit les coûts en mettant en cache le contenu fréquemment consulté. Amazon CloudFront peut distribuer votre contenu dans le monde entier en le mettant en cache et en le diffusant au plus près de vos joueurs, tout en réduisant la charge de téléchargement de votre jeu, comme Amazon S3.

Plutôt que de diffuser votre contenu public directement à partir de compartiments S3, il est recommandé de garder ce contenu privé et de le diffuser publiquement en utilisant CloudFront. CloudFront peut être configuré pour obliger les joueurs à accéder à votre contenu privé (tel que le téléchargement d'un nouveau jeu pour les joueurs payants uniquement) en utilisant des cookies signés URLs ou signés. Vous pouvez ensuite développer votre application soit pour créer et distribuer des cookies signés URLs aux utilisateurs authentifiés, soit pour envoyer des en-têtes set-cookie qui définissent des cookies signés pour les utilisateurs authentifiés. Lorsque vous créez des cookies

signés URLs ou signés pour contrôler l'accès à vos fichiers, vous pouvez spécifier une date et une heure de fin, après quoi l'URL et les cookies ne sont plus valides.

En option, vous pouvez également spécifier l'adresse IP ou la plage d'adresses des ordinateurs qui peuvent être utilisés pour accéder à votre contenu, ce qui est utile si vous souhaitez restreindre l'accès à des partenaires de studios de développement de jeux ou à des réseaux de sous-traitants spécifiques. Utilisez des cookies signés lorsque vous souhaitez donner accès à plusieurs fichiers restreints ou si vous ne souhaitez pas modifier votre fichier actuel URLs. Utilisez URLs Signed lorsque vous souhaitez restreindre l'accès à des fichiers individuels ou si vos utilisateurs utilisent un client qui ne prend pas en charge les cookies. Les cookies URLs signés ont priorité sur les cookies signés.

Étapes d’implémentation

- Utilisez les rôles IAM et les politiques de compartiment pour accorder un accès approprié aux compartiments S3 pour les applications internes, les équipes ou les scénarios entre comptes.
- Générez des fichiers présignés URLs pour accorder un accès à court terme aux objets S3, adaptés au contenu téléchargeable ou aux téléchargements temporaires tels que les journaux des clients.
- Utilisez Amazon CloudFront avec des cookies URLs ou des cookies pour proposer du contenu privé aux utilisateurs authentifiés de manière plus sécurisée

GAMESEC04-BP02 Limiter l'accès d'origine aux réseaux de diffusion de contenu autorisés () CDNs

Empêchez les utilisateurs de contourner vos réseaux de diffusion de contenu pour accéder directement au contenu provenant de votre origine, tel que vos compartiments Amazon S3. Il est important de limiter l'accès à votre source aux seules personnes autorisées CDNs, afin de réduire les coûts de transfert de données liés à la diffusion inutilement de contenu hors de l'origine. Il améliore également votre niveau de sécurité en faisant passer l'accès public à votre contenu d'origine via le même point d'entrée, où vous pouvez déployer des contrôles de sécurité périphériques tels que le filtrage de AWS WAF couche 7, l'injection et l'inspection des paramètres de requête HTTP liés à la sécurité, ainsi que des protections par déni de service (DDoS) distribué.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Pour implémenter ces contrôles pour une origine Amazon S3, vous pouvez utiliser une identité CloudFront d'accès à l'origine (OAI) Amazon, qui vérifie que les demandes adressées à vos objets S3 proviennent de votre CloudFront distribution. Associez-le AWS WAF à votre CloudFront distribution pour fournir un filtrage de couche 7. Toutefois, si vous diffusez du contenu supplémentaire CDNs, vous pouvez configurer le CDN pour insérer un ou plusieurs en-têtes HTTP personnalisés dans les demandes d'origine, qui peuvent être inspectés AWS WAF pour vérifier que le trafic entrant provient de votre fournisseur de CDN autorisé.

Cette approche est également utile pour empêcher les utilisateurs de contourner vos fournisseurs de CDN lorsque votre origine est hébergée derrière un Application Load Balancer (ALB). ALBs peut être associé AWS WAF pour les protections de couche 7. Vous pouvez configurer AWS WAF pour insérer un en-tête HTTP personnalisé qui sera inspecté par votre ALB afin de traiter et d'inspecter le trafic entrant vers l'équilibreur de charge par. AWS WAF

Exemple client

AnyCompany Les jeux mettent en œuvre des restrictions d'accès à Origin pour protéger leurs actifs de jeu, leur contenu téléchargeable et leurs fichiers correctifs contre tout accès direct non autorisé qui pourrait permettre aux joueurs de contourner les contrôles de sécurité ou d'obtenir du contenu premium sans authentification appropriée. Cette approche leur permet de surveiller les modèles d'accès au contenu via un point centralisé, ce qui leur permet d'identifier facilement les comportements de téléchargement suspects susceptibles d'indiquer la présence d'attaques coordonnées ou de redistribution de contenu non autorisée.

Étapes d’implémentation

- Utilisez Amazon CloudFront Origin Access Identity (OAI) pour restreindre l'accès direct aux objets S3
- AWS WAF Associez-le à CloudFront ou ALB pour fournir un filtrage de couche 7 et contribuer à la protection contre les attaques DDo S et les requêtes malveillantes.
- Configurez des en-têtes HTTP personnalisés dans Cloudfront pour vérifier que le trafic entrant provient de sources autorisées.

GAMESEC04-BP03 Implémenter des restrictions géographiques pour limiter les accès non autorisés

Lorsqu'un joueur demande votre contenu, Amazon CloudFront diffuse le contenu demandé depuis l'emplacement périphérique le plus proche, quel que soit l'endroit où se trouve le lecteur. Cependant, il peut arriver que vous deviez restreindre la façon dont votre contenu est accessible aux utilisateurs dans certaines régions du monde. Par exemple, vous pouvez avoir une stratégie de déploiement de jeux continue qui publie le contenu par étapes, ou vous pouvez avoir à respecter des contrôles d'accès spécifiques à chaque pays. country-by-country

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Vous pouvez utiliser des restrictions géographiques, également appelées blocage géographique, pour empêcher les joueurs situés dans des zones géographiques spécifiques d'accéder au contenu que vous distribuez par le biais d'une CloudFront distribution. Cette fonctionnalité vous permet de restreindre l'accès aux fichiers associés à une distribution et de restreindre l'accès au niveau du pays. Vous pouvez également utiliser un service de géolocalisation tiers pour restreindre l'accès à un sous-ensemble des fichiers associés à une distribution ou pour restreindre l'accès à une granularité plus fine qu'au niveau du pays.

En utilisant des restrictions CloudFront géographiques, vous pouvez autoriser vos joueurs à accéder à votre contenu uniquement s'ils se trouvent dans l'un des pays autorisés figurant sur la liste des pays autorisés. Vous pouvez également empêcher vos joueurs d'accéder à votre contenu s'ils se trouvent dans l'un des pays interdits. Si une demande provient d'un emplacement géographique bloqué, le joueur CloudFront recevra un code d'état HTTP 403 Forbidden. Il est important de noter que cela fonctionne bien pour le contenu non sensible et ne doit pas être utilisé comme protection autonome pour les informations personnelles ou les artefacts de jeu sensibles.

Étapes d’implémentation

- Utilisez les restrictions CloudFront géographiques pour autoriser ou refuser l'accès au contenu en fonction des listes d'autorisation ou de refus au niveau du pays.
- Renvoie un code d'état HTTP 403 Forbidden pour les demandes provenant de zones géographiques bloquées.
- Évitez de vous fier uniquement aux restrictions géographiques pour protéger le contenu sensible ou les informations personnelles

GAMESEC04-BP04 Restreignez l'accès au contenu avec des solutions de gestion des droits numériques (DRM)

Envisagez de restreindre l'accès au contenu de votre jeu en utilisant des outils de chiffrement puissants tels qu'une solution de gestion des droits numériques (DRM). Ce type de solution peut être utilisé pour chiffrer votre contenu privé et distribuer les clés de déchiffrement aux joueurs autorisés.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Les solutions DRM sont recommandées dans les situations où vous souhaitez autoriser les joueurs à télécharger le contenu du jeu plus tôt, mais vous ne voulez pas qu'ils puissent accéder au contenu ou y jouer avant une heure prédéterminée. Par exemple, cela est courant dans les situations où les joueurs sont autorisés à précommander un jeu et à configurer leur client de jeu pour qu'il commence automatiquement à télécharger les fichiers cryptés de manière anticipée. Cette stratégie vérifie que le jeu est téléchargé et prêt à être joué une fois le jeu officiellement sorti. Après la sortie du jeu, le client du jeu du joueur peut demander des clés de déchiffrement à la solution principale DRM afin de déchiffrer les fichiers précédemment téléchargés et de commencer à jouer au jeu.

Les systèmes DRM sont également utilisés pour bloquer la redistribution et la manipulation non autorisées de jeux après leur téléchargement et leur installation par un joueur autorisé. Les systèmes DRM nécessitent une intégration avec l'origine pour échanger des clés de chiffrement et autoriser les joueurs à récupérer la clé de déchiffrement. Les fournisseurs commerciaux de DRM proposent une gamme de solutions dotées de fonctionnalités et d'un support pour différents appareils.

Étapes d’implémentation

- Utilisez des solutions DRM pour chiffrer le contenu privé des jeux et distribuer des clés de déchiffrement aux joueurs autorisés.
- Activez le prétéléchargement de fichiers cryptés pour les jeux précommandés, en débloquant l'accès à l'aide de clés de déchiffrement au moment de la sortie.
- Intégrez les systèmes DRM à l'origine pour gérer les clés de chiffrement et bloquer la redistribution ou la manipulation non autorisées du contenu.

Détection

GAMESEC05 : Comment surveillez-vous et analysez-vous le comportement d'utilisation des joueurs dans votre jeu ?

La surveillance et l'analyse du comportement d'utilisation des joueurs sont essentielles pour les studios de jeux, car elles vous permettent de détecter les menaces de sécurité, les tricheries et autres formes de comportement abusif susceptibles de compromettre l'intégrité du jeu et la sécurité des joueurs. En suivant des modèles tels que des taux de progression inhabituels, des transactions anormales en jeu ou des comportements de communication suspects, vous pouvez identifier les tricheurs potentiels, les comptes frauduleux ou les menaces coordonnées avant qu'ils n'aient un impact significatif sur l'expérience des joueurs.

Bonnes pratiques

- GAMESEC05-BP01 Mettre en œuvre une stratégie complète de collecte de données pour surveiller le comportement des joueurs
- GAMESEC05-BP02 Collecter, stocker et analyser les journaux d'utilisation des joueurs pour détecter les comportements inappropriés

GAMESEC05-BP01 Mettre en œuvre une stratégie complète de collecte de données pour surveiller le comportement des joueurs

Pour maintenir une expérience positive pour les joueurs, mettez en œuvre une stratégie complète de collecte et d'analyse des données. La capture, le stockage et l'analyse des données pertinentes fournissent des informations sur la façon dont les joueurs interagissent avec les fonctionnalités de votre jeu et entre eux. Cette approche axée sur les données peut guider la prise de décision, améliorer l'engagement et la fidélisation des joueurs, optimiser les stratégies de monétisation et, en fin de compte, améliorer l'expérience globale des joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Mettez en œuvre des systèmes de collecte de données pour capturer et enregistrer les actions pertinentes des joueurs, telles que les sessions de jeu, les progrès, les succès, les achats,

les interactions avec les éléments du jeu et les activités sociales. Collectez des données côté serveur telles que la charge du serveur, le trafic réseau et les journaux d'erreurs pour surveiller les performances techniques et identifier les problèmes potentiels. Recueillez les commentaires des joueurs par le biais de sondages, de forums, de tickets d'assistance et de réseaux sociaux pour comprendre leurs expériences et leurs préférences.

Lorsque vous stockez vos données de jeu, établissez un entrepôt de données centralisé ou un lac de données pour stocker et organiser les données collectées et implémentez des pipelines pour le nettoyage, la transformation et l'agrégation des données afin de préparer les données pour une analyse efficace.

Après avoir stocké les données, analysez-les pour obtenir des informations telles que la rétention et le taux de désabonnement des joueurs, les stratégies de monétisation et l'utilisation des fonctionnalités grâce à des outils de visualisation des données.

Étapes d’implémentation

- Capturez et enregistrez les actions des joueurs, les indicateurs côté serveur et les commentaires pour surveiller les interactions et les performances techniques.
- Utilisez un entrepôt de données centralisé tel qu'Amazon Redshift ou le lac de données S3 pour stocker, nettoyer, transformer et organiser les données de jeu à des fins d'analyse.
- Analysez les données collectées à l'aide d'outils de visualisation, tels qu'Amazon Quicksight, pour mieux comprendre la fidélisation des joueurs, la monétisation et l'utilisation des fonctionnalités.

GAMESEC05-BP02 Collecter, stocker et analyser les journaux d'utilisation des joueurs pour détecter les comportements inappropriés

Utilisez votre jeu pour collecter des journaux afin de comprendre comment les joueurs utilisent les fonctionnalités de votre jeu et comment ils interagissent avec les autres joueurs. Vous pouvez ensuite bloquer les activités non autorisées susceptibles de dégrader l'expérience des joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Envoyez des événements de log structurés au Game Analytics Pipeline, en utilisant une solution de journalisation telle qu'Amazon CloudWatch Logs ou Amazon OpenSearch Service, ou via une solution d'un AWS partenaire tel que Datadog, Sumo Logic, New Relic, Honeycomb.io ou Splunk.

Structurez ces journaux d'utilisation des joueurs afin qu'ils puissent être utilisés pour détecter les cas où des actions spécifiques des joueurs doivent être étudiées.

Après avoir capturé les données, envisagez de mettre en œuvre des outils pour détecter les comportements d'utilisation inappropriés. Par exemple, si votre jeu comporte des fonctionnalités sociales telles que la messagerie intégrée aux joueurs, le chat vocal ou les forums en ligne, enregistrez les journaux de ces engagements des joueurs dans un format qui peut être analysé à des fins de modération.

Configurez la fonction de chat vocal de votre jeu pour exporter les enregistrements vers Amazon S3 et utilisez Amazon Transcribe pour convertir le discours audio au format texte qui peut être stocké pour traitement. Vous pouvez également effectuer une transcription en streaming en temps réel en intégrant le service de chat vocal de votre jeu directement à l'API Transcribe pour transcrire le streaming audio en temps réel. Les équipes de modération peuvent examiner le contenu manuellement, et une fois que le contenu est dans un format standard, vous pouvez également utiliser les services d'intelligence AWS artificielle et d'apprentissage automatique pour effectuer la modération automatiquement. Amazon Comprehend peut être utilisé pour effectuer un traitement du langage naturel (NLP) afin de découvrir des informations contenues dans le texte non structuré, ce qui permet de classer et d'organiser les conversations en sujets pertinents et d'identifier les comportements inappropriés tels que les grossièretés.

Étapes d’implémentation

- Collectez, stockez et analysez les journaux d'utilisation des joueurs.
- Utilisez les AWS services d'intelligence artificielle et d'apprentissage automatique pour examiner et analyser plus efficacement les journaux d'utilisation de vos joueurs.

Protection de l’infrastructure

Consultez le livre blanc Well-Architected Framework pour connaître les meilleures pratiques en matière de protection de l'infrastructure en matière de sécurité applicables aux charges de travail des jeux.

GAMESEC06 : Comment surveillez-vous les menaces liées à l'infrastructure et y répondez-vous ?

La surveillance et la réponse aux menaces liées à l'infrastructure sont essentielles pour les studios de jeux vidéo, car leur infrastructure constitue l'épine dorsale qui prend en charge des millions de joueurs simultanés, traite les transactions en argent réel et stocke des données précieuses sur les joueurs ainsi que du contenu de jeu propriétaire. Il est essentiel que les studios de jeux mettent en œuvre des systèmes de surveillance et des processus de réponse aux incidents capables de préserver l'intégrité des expériences des joueurs et des opérations commerciales.

Bonnes pratiques

- GAMESEC06-BP01 Utilisez des outils pour détecter et répondre aux menaces qui pèsent sur votre infrastructure
- GAMESEC06-BP02 Utilisez l'intelligence artificielle et les outils d'apprentissage automatique pour automatiser certains aspects de votre stratégie de protection de l'infrastructure
- GAMESEC06-BP03 Utilisez les informations issues des journaux au niveau du système pour améliorer en permanence votre stratégie de protection de l'infrastructure

GAMESEC06-BP01 Utilisez des outils pour détecter et répondre aux menaces qui pèsent sur votre infrastructure

Pour surveiller en permanence les activités malveillantes et les comportements non autorisés au sein de votre AWS environnement, pensez à utiliser Amazon GuardDuty. GuardDuty identifie les menaces en surveillant le comportement des comptes, l'activité du réseau et les modèles d'accès aux données au sein de votre environnement. Il analyse les événements issus de plusieurs sources de données, telles que les journaux d' CloudTrail événements, les journaux de flux Amazon VPC et les journaux DNS pour détecter les menaces potentielles. En intégrant Amazon CloudWatch Events et Lambda, les GuardDuty alertes peuvent être automatiquement transmises aux équipes de sécurité concernées pour une analyse plus approfondie.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

AWS Security Hub CSPMfournit une vue complète de l'état de votre sécurité AWS et vérifie que votre environnement est conforme aux normes et aux meilleures pratiques du secteur de la sécurité. Security Hub CSPM collecte des données de sécurité provenant de l'ensemble Comptes AWS des services et des produits partenaires tiers pris en charge, analyse vos tendances en matière de sécurité et identifie les problèmes de sécurité les plus prioritaires. L' GuardDutyintégration d'Amazon

à Security Hub CSPM vous permet d'envoyer des résultats depuis GuardDuty Security Hub CSPM. Security Hub CSPM peut ensuite inclure ces résultats dans son analyse de votre posture de sécurité.

Il est courant que les mauvais acteurs utilisent des robots pour prendre le contrôle de comptes et tricher dans les jeux. WAF Bot Control vous donne une visibilité et un contrôle sur le trafic de bots courant et omniprésent qui peut consommer des ressources excédentaires, fausser les indicateurs, provoquer des interruptions de service ou effectuer d'autres activités indésirables.

Un rançongiciel est un code malveillant conçu pour obtenir un accès non autorisé aux systèmes et aux ensembles de données et pour chiffrer ces données afin de bloquer l'accès des joueurs légitimes. Une fois que le ransomware a bloqué l'accès des joueurs à leurs systèmes et chiffré leurs données sensibles, les cybercriminels exigent une rançon avant de fournir une clé de déchiffrement pour déverrouiller les données. Organisations peuvent être complètement fermées à la suite d'un événement malveillant, ce qui entraîne des coûts importants et une perte de productivité. Reportez-vous à la section Sécurisation de votre AWS Cloud environnement contre les ransomwares pour connaître les meilleures pratiques que vous pouvez appliquer pour renforcer votre capacité à lutter contre les ransomwares avant, pendant et après un incident.

Votre jeu peut permettre aux joueurs de contacter des agents d'assistance aux joueurs par le biais d'un centre d'appels tel qu'Amazon Connect ou de chatbots utilisant Amazon Lex. Amazon Connect fournit une assistance pour surveiller les conversations en direct et enregistrées. Pour analyser les interactions entre les joueurs et les robots de discussion d'assistance aux joueurs créés avec Amazon Lex, vous pouvez stocker les journaux de conversation issus de ces interactions dans Amazon CloudWatch Logs, qui peuvent être exportés vers Amazon S3 et analysés comme décrit précédemment.

Enfin, effectuez des tests de pénétration dans le cadre de votre stratégie de protection de l'infrastructure. Que vous réalisiez ces évaluations en interne ou par l'intermédiaire d'un AWS partenaire, respectez les politiques de support AWS client relatives aux tests d'intrusion.

Étapes d’implémentation

- Utilisez Amazon GuardDuty pour surveiller le comportement des comptes, l'activité du réseau et les modèles d'accès aux données afin de détecter les menaces, et intégrez le Security Hub CSPM pour obtenir une vue unifiée de la sécurité.
- Mettez en œuvre le contrôle des AWS WAF bots pour détecter et atténuer le trafic de bots susceptible de nuire aux ressources et à l'expérience des joueurs.
- Effectuez régulièrement des tests d'intrusion, conformément aux politiques de support AWS client, afin d'évaluer et de renforcer votre posture de sécurité.

GAMESEC06-BP02 Utilisez l'intelligence artificielle et les outils d'apprentissage automatique pour automatiser certains aspects de votre stratégie de protection de l'infrastructure

Amazon Lookout for Metrics utilise le machine learning pour détecter et diagnostiquer automatiquement les anomalies dans vos données commerciales et opérationnelles et surveille les indicateurs les plus importants pour votre entreprise avec une rapidité et une précision accrues. Le service permet également de diagnostiquer facilement la cause première des anomalies, telles qu'une baisse soudaine des revenus, des connexions, des transactions ou de la rétention. Il n'est pas nécessaire que les développeurs de jeux aient une expérience du machine learning pour le configurer et puissent se connecter à des sources de données populaires, notamment Amazon S3, Amazon CloudWatch, Amazon RDS, Amazon Redshift, ainsi qu'à de nombreuses applications SaaS. Par exemple, vous pouvez intégrer Amazon Lookout for Metrics au Game Analytics Pipeline et à d'autres sources de données pour commencer à analyser le comportement afin de détecter les anomalies.

Niveau d'exposition au risque si cette bonne pratique n'est pas respectée : élevé

Directives d'implémentation

Vous pouvez également choisir de créer, de former et d'héberger un modèle d'apprentissage automatique personnalisé à l'aide d'Amazon SageMaker AI AI pour traiter des cas d'utilisation tels que la modération du contenu, la détection de toxicité, la détection de triche, la détection de fraude, etc.

Exemple client

AnyCompany Games utilise Amazon Lookout for Metrics pour détecter automatiquement les tendances inhabituelles dans les performances du serveur, les tentatives de connexion des joueurs ou les volumes de transactions susceptibles d'indiquer des menaces émanant de personnes mal intentionnées. En outre, ils ont utilisé Amazon SageMaker AI pour développer des modèles d'apprentissage automatique personnalisés qui analysent en permanence les modèles de trafic réseau et le comportement des joueurs afin d'identifier les menaces coordonnées, telles que les réseaux de robots qui tentent d'exploiter leur économie virtuelle.

Cette approche automatisée permet à leur équipe de sécurité de se concentrer sur l'investigation et la réponse aux menaces réelles plutôt que de surveiller manuellement des milliers de métriques, tout en s'assurant que les modèles de menaces émergents sont détectés et corrigés avant qu'ils n'aient un impact significatif sur la disponibilité du jeu ou la sécurité des joueurs.

Étapes d’implémentation

- Utilisez Amazon Lookout for Metrics pour détecter et diagnostiquer automatiquement les anomalies dans les données commerciales et opérationnelles clés
- Intégrez Amazon Lookout for Metrics à des sources de données telles que Game Analytics Pipeline, Amazon S3, CloudWatch ou pour surveiller des indicateurs tels que les revenus, les connexions et la rétention.
- Utilisez Amazon SageMaker AI pour créer, former et héberger des modèles d'apprentissage automatique personnalisés pour des cas d'utilisation avancés tels que la détection des triches, la prévention des fraudes et la modération du contenu.

GAMESEC06-BP03 Utilisez les informations issues des journaux au niveau du système pour améliorer en permanence votre stratégie de protection de l'infrastructure

Capturez et stockez les journaux au niveau du système à partir des services pertinents, tels que les journaux d'accès au serveur S3, les journaux CloudFrontd'accès et les journaux d'accès ALB. Ces journaux peuvent être stockés dans un compartiment S3 de votre compte et sont utiles pour associer les informations relatives à l'utilisation que vous faites des joueurs depuis le jeu à des informations au niveau du système, notamment les détails de connexion tels que les adresses IP, les en-têtes de requêtes, ainsi que le traitement et le filtrage des demandes pertinents que vous avez peut-être configurés dans le backend de votre jeu. Vous pouvez envoyer ces journaux aux mêmes solutions de journalisation mentionnées précédemment, et vous pouvez les analyser à l'aide de requêtes SQL avec Amazon Athena sans avoir à déplacer les journaux hors d'Amazon S3.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Access Analyzer for S3 est une fonctionnalité qui surveille vos politiques d'accès aux compartiments, en s'assurant qu'elles fournissent uniquement l'accès prévu à vos ressources Amazon S3. Access Analyzer for S3 évalue vos politiques d'accès aux compartiments et vous permet de détecter et de corriger rapidement les compartiments présentant un accès potentiellement involontaire.

Étapes d’implémentation

- Utilisez les AWS services de détection des menaces et de réponse aux incidents afin d'automatiser certains aspects de votre stratégie de protection de l'infrastructure.

- Obtenez des informations sur la protection de votre infrastructure grâce à des journaux au niveau du système et à AWS des services d'intelligence artificielle et d'apprentissage automatique.

Protection des données

Lors du développement et de l'architecture de votre jeu, réfléchissez au type de données que votre studio collecte et à la manière dont vous avez décidé de les protéger. Les sujets à explorer dans le cadre de cet aspect de la sécurité incluent :

- Comment vous avez choisi d'identifier et de classer vos données
- Comment protégez-vous les données au repos
- Comment protégez-vous les données en transit

Il n'existe aucune bonne pratique en matière de protection des données spécifique au Games Lens. Consultez le livre blanc Well-Architected Framework pour connaître les meilleures pratiques en matière de protection des données à des fins de sécurité.

Intervention en cas d’incidents

GAMESEC07 : Comment définissez-vous et appliquez-vous les politiques visant à répondre aux inconduites et aux comportements abusifs des joueurs ?

L'inconduite et les comportements abusifs des joueurs peuvent avoir un impact significatif sur l'expérience de vos joueurs. Le mauvais comportement des joueurs peut faire fuir les joueurs légitimes, ce qui peut entraîner une baisse de la rétention des joueurs, une baisse des revenus provenant des achats en jeu et des critiques négatives susceptibles de nuire à la réputation d'un jeu et à ses ventes futures.

Définissez des politiques qui encouragent les actions positives auprès de vos joueurs et déterminez comment vous allez appliquer ces politiques.

Bonnes pratiques

- GAMESEC07-BP01 Mettre en œuvre un plan de réponse aux incidents pour gérer les mauvais acteurs et les comportements abusifs
- GAMESEC07-BP02 Bannir les comptes associés à de mauvais acteurs

GAMESEC07-BP01 Mettre en œuvre un plan de réponse aux incidents pour gérer les mauvais acteurs et les comportements abusifs

Créez un plan d'action pour répondre aux mauvais acteurs et aux comportements abusifs dans votre jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Tenez compte de facteurs tels que le moment de suspendre temporairement ou de bannir définitivement les joueurs et la durée pendant laquelle les informations d'identification des joueurs temporairement suspendus doivent être désactivées.

Exemple client

AnyCompany Games crée un système de réponse aux incidents à plusieurs niveaux dans lequel les infractions mineures, telles que les messages de chat inappropriés, entraînent la suspension automatique du compte pendant 24 heures, tandis que les violations plus graves telles que la tricherie ou le harcèlement entraînent des suspensions immédiates de 7 jours avec un examen obligatoire par des modérateurs humains.

De plus, les AnyCompany Jeux établissent des procédures d'escalade dans le cadre desquelles les récidivistes sont passibles de suspensions de plus en plus longues. Ils créent des processus d'appel qui permettent aux joueurs faussement signalés de contester des actions automatisées tout en maintenant la sécurité grâce à des exigences de vérification d'identité.

GAMESEC07-BP02 Bannir les comptes associés à de mauvais acteurs

S'ils ne sont pas atténués, les comportements abusifs dans un jeu peuvent continuer à avoir un impact négatif sur l'expérience de jeu des autres joueurs et doivent être atténués dès que possible. Mettez en place un processus visant à imposer des interdictions ou d'autres formes de restrictions aux acteurs malveillants dont il est confirmé qu'ils enfreignent vos conditions d'utilisation.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Généralement, les règles et le processus d'évaluation permettant de déterminer les circonstances dans lesquelles ce type de restrictions est imposé seront déterminés par le personnel, tel qu'une équipe de la communauté de joueurs ou une équipe chargée de la confiance et de la sécurité au

sein de votre organisation. Après avoir signalé les mauvais acteurs, exécutez un flux de travail prédéterminé pour agir sur les joueurs identifiés.

Par exemple, AWS Lambdales fonctions AWS Step Functionset peuvent être utilisées pour exécuter un flux de travail automatisé qui accepte un lot de comptes de joueurs en entrée. Le flux de travail met ensuite à jour les entrées d'une table Amazon DynamoDB appelée Bans, qui peut inclure des informations sur le compte du joueur, le motif et la durée du bannissement.

En fonction de la conception de votre jeu et de votre système de gestion de compte et du type d'abus que vous rencontrez de la part d'acteurs malveillants, maintenez un système d'enregistrement des bannissements distinct de votre système de gestion de compte. Il se peut que vous ne souhaitiez pas désactiver le compte du joueur dans votre système de gestion de compte, mais plutôt simplement le désactiver pour qu'il puisse jouer à votre jeu. Cela peut être utile dans les situations où les informations d'identification du compte du joueur sont utilisées pour accéder à plusieurs jeux avec des conditions d'utilisation ou des politiques différentes.

Étapes d’implémentation

- Définissez et appliquez des politiques pour répondre aux comportements abusifs de la part de mauvais acteurs.
- Utilisez AWS les services pour automatiser vos réponses aux acteurs malveillants.

Ressources

- AWS Guide technique de réponse aux incidents de sécurité
- AWS Blog Machine Learning : Détectez les utilisateurs réels et réels et dissuadez les acteurs malveillants grâce à Amazon Rekognition Face Liveness
- AWS Solutions pour les jeux : Community Health

Sécurité des applications

GAMESEC08 : Comment sécurisez-vous votre pipeline ? CI/CD

Un CI/CD pipeline de développement de jeux est généralement composé de serveurs de contrôle de source et de stockage à haute disponibilité, de ressources informatiques pour exécuter vos builds

et de logiciels pour effectuer des tests automatisés, ainsi que de la connectivité réseau appropriée depuis vos machines de développement. La sécurisation de votre CI/CD pipeline est importante pour protéger les informations sensibles, préserver l'intégrité du code et garantir la fiabilité des versions. L'intégration de la gouvernance et des garde-fous permet aux développeurs de gagner en agilité tout en maintenant les bonnes pratiques de sécurité.

Étant donné que les jeux gèrent souvent le traitement des paiements, stockent des informations personnelles et maintiennent des économies virtuelles valant de l'argent réel, une faille de sécurité dans le processus de développement peut entraîner des pertes financières importantes, des sanctions réglementaires et une perte de confiance des joueurs.

En intégrant des mesures de protection, les entreprises conservent la visibilité et le contrôle du processus de livraison des logiciels, ce qui permet de réagir rapidement aux incidents et de promouvoir une culture de pratiques de codage sécurisées.

Bonnes pratiques

- GAMESEC08-BP01 Appliquer la sécurité à chaque étape du pipeline CI/CD

GAMESEC08-BP01 Appliquer la sécurité à chaque étape du pipeline CI/CD

Les garde-fous tels que les contrôles d'accès, la séparation des tâches et les pistes d'audit fournissent une protection contre les accès non autorisés ou les activités malveillantes.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Votre personnel, vos processus et votre technologie devraient également sécuriser le pipeline. Les personnes les plus proches du code doivent établir des pratiques de codage sécurisées et s'assurer qu'elles les respectent. Effectuez des itérations continues sur vos processus pour vérifier que le niveau de sécurité est constant tout au long du pipeline. Enfin, mettez en œuvre une technologie pour vérifier que les meilleures pratiques et processus ne sont pas contournés.

Exemple client

AnyCompany Games met en œuvre des contrôles d'accès basés sur les rôles dans lesquels seuls les développeurs expérimentés peuvent approuver les modifications apportées au code de leur système anti-triche, tout en obligeant les membres de l'équipe de sécurité à réviser le code pour les composants qui traitent les données de paiement des joueurs.

Leur CI/CD pipeline exécute automatiquement des contrôles de validation des modèles de menaces, afin de s'assurer que les nouvelles fonctionnalités, telles qu'une place de marché réservée aux joueurs, sont testées contre des vecteurs d'attaque préalablement identifiés, tels que les exploits de duplication d'objets ou les tentatives de transactions frauduleuses.

Étapes d’implémentation

- Fournissez aux utilisateurs des autorisations sur la base du principe du moindre privilège.
- AWS CloudTrail À utiliser pour auditer les appels d'API effectués entre les services utilisés dans le pipeline.
- Utilisez des hooks de pré-validation pour vérifier que le code respecte les pratiques générales et les politiques de l'entreprise.

Automatisez la sécurité

GAMESEC09 : Comment automatisez-vous la sécurité au sein de votre pipeline ? CI/CD

Intégrez des mesures de sécurité dans votre CI/CD pipeline afin de maintenir une posture de sécurité robuste tout au long du cycle de développement. Ce processus offre bon nombre des mêmes avantages que la sécurité de votre pipeline. Le fait de disposer d'un CI/CD pipeline sécurisé réduit le risque d'événements de sécurité susceptibles de retarder le calendrier de développement du jeu.

Pour garantir la sécurité de votre CI/CD pipeline, vous devez mettre en œuvre les meilleures pratiques et outils de sécurité à chaque étape du cycle de développement. L'intégration de la sécurité vous permet également de réduire le temps consacré aux examens de sécurité.

L'automatisation de la sécurité au sein de votre CI/CD pipeline est particulièrement cruciale pour vérifier que les contrôles de sécurité sont systématiquement mis en œuvre et testés à chaque modification du code. La mise en œuvre des outils et de l'automatisation appropriés peut fournir des jeux sûrs et sécurisés.

Bonnes pratiques

- GAMESEC09-BP01 Intégrez l'outillage et l'automatisation pour réduire le temps moyen des révisions de sécurité

GAMESEC09-BP01 Intégrez l'outillage et l'automatisation pour réduire le temps moyen des révisions de sécurité

Pour identifier les failles de sécurité, les entreprises peuvent utiliser différents outils et services tels que les tests statiques de sécurité des applications (SAST) et les tests dynamiques de sécurité des applications (DAST). Le SAST est un moyen d'examiner le code source et de déterminer les failles de sécurité. DAST est une méthode de type boîte noire qui permet de tester votre code en testant vos applications sans consulter le code source.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Un autre outil que les entreprises peuvent utiliser est l'analyse de la composition logicielle (SCA), qui évalue la sécurité de vos dépendances tierces ou open source. Pour une approche plus manuelle, des révisions de code sécurisées peuvent être mises en œuvre tout au long du pipeline.

Exemple client

AnyCompany Games utilise les outils SAST pour signaler automatiquement les failles de sécurité potentielles au cours du processus de développement. Ils utilisent également les outils DAST pour simuler les menaces pesant sur les versions de jeu en cours d'exécution afin de vérifier que les contrôles de sécurité fonctionnent comme prévu. De plus, AnyCompany Games intègre des outils d'analyse des dépendances dans son processus de développement afin d'identifier automatiquement les vulnérabilités connues dans les bibliothèques et les moteurs de jeu tiers.

Étapes d’implémentation

- Utilisez Amazon CodeGuru comme outil SAST.
- Utilisez des outils open source tels que OWASP Dependency Check, SonarQube ou. OWASPZap

Ressources

- Sécurité pour les développeurs

Modélisation des menaces

GAMESEC10 : Comment intégrez-vous la modélisation des menaces dans le cycle de développement des applications de votre entreprise ?

La modélisation des menaces est le processus qui consiste à identifier et à hiérarchiser les menaces potentielles pour votre application et à déterminer les solutions qui peuvent être utilisées pour les atténuer. Cette pratique est devenue de plus en plus importante à mesure que les jeux sont devenus des systèmes complexes et connectés qui gèrent les données sensibles des utilisateurs et les transactions en argent réel.

Intégrez la modélisation des menaces en tant qu'exercice continu pour renforcer la sécurité du jeu, non seulement lors de la phase de conception initiale, mais aussi au fur et à mesure que le jeu continue de croître et d'évoluer.

Bonnes pratiques

- GAMESEC10-BP01 Déterminez quand et comment effectuer des exercices de modélisation des menaces tout au long du cycle de développement de vos applications

GAMESEC10-BP01 Déterminez quand et comment effectuer des exercices de modélisation des menaces tout au long du cycle de développement de vos applications

Il n'existe pas de meilleure façon d'aborder la modélisation des menaces. Les détails indiquant quand et comment procéder varient en fonction des besoins uniques de votre studio de jeu. Par exemple, selon la taille de votre studio, certains membres de votre équipe peuvent être impliqués dans un ou plusieurs aspects du processus de modélisation des menaces.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Le blog sur la AWS sécurité fournit une vue d'ensemble des points à prendre en compte lors de l'élaboration de votre stratégie de modélisation des menaces, tels que :

- Quels membres et personnalités de votre équipe devraient participer à la modélisation des menaces
- Comment déterminer les outils de flux de travail appropriés à utiliser
- Comment déterminer la propriété des différents aspects de la modélisation des menaces
- Comment identifier et évaluer les contrôles de sécurité à utiliser dans le cadre de la conception de votre charge de travail

Exemple client

AnyCompany Les jeux commencent par cataloguer des actifs précieux tels que les données des joueurs, le code et les algorithmes du jeu, les devises du jeu, le contenu généré par les utilisateurs et les propriétés intellectuelles telles que le contenu inédit ou les moteurs propriétaires. Ils prennent en compte différents types de mauvais acteurs potentiels, tels que les tricheurs recherchant des avantages injustes, les mauvais acteurs qui tentent de voler des données personnelles ou financières et les utilisateurs malveillants qui tentent de perturber le jeu.

Tout au long du processus de développement, AnyCompany Games utilise des modèles de menaces pour orienter les pratiques de codage sécurisé et influencer les stratégies de test afin de se concentrer sur les zones à haut risque. Avant le lancement d'un jeu, ils procèdent à des examens complets de la modélisation des menaces afin d'évaluer l'état de préparation aux chargements de joueurs prévus et aux tentatives d'accès non autorisées, et de préparer les procédures de réponse aux incidents.

Étapes d’implémentation

- Mettez en place des garde-corps à chaque étape de votre CI/CD pipeline.
- Utilisez l'automatisation et les outils pour améliorer l'efficacité des examens de sécurité de vos applications.
- Utilisez la modélisation des menaces comme processus pour améliorer la sécurité de vos applications.

Ressources

- AWS Blog sur la sécurité : Comment aborder la modélisation des menaces
- NIST : Guide de modélisation des menaces systémiques centrées sur les données

- Modéliser les menaces de la bonne manière pour les constructeurs : formation virtuelle à suivre à leur rythme avec AWS Skill Builder
- Modélisation des menaces pour les constructeurs — AWS Atelier

Ressources

Consultez les ressources suivantes pour en savoir plus sur nos meilleures pratiques en matière de sécurité.

Documents connexes :

- Scénarios Amazon Cognito courants
- Utilisation de documents signés URLs
- Utilisez les flux de canaux pour supprimer le contenu grossier et sensible des messages dans la messagerie du SDK Amazon Chime
- Sécurité sur Amazon GameLift
- Diffusion de contenu sécurisée via Amazon CloudFront
- Guide de réponse en matière de sécurité
- AWS Meilleures pratiques pour la résilience DDo des systèmes
- Sécurisation de votre AWS Cloud environnement contre les ransomwares

Solutions partenaires associées :

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

Supports de formation connexes :

- Commencer à utiliser Amazon Cognito
- Formation à votre propre rythme en matière de sécurité

Fiabilité

Le pilier de fiabilité inclut la capacité d'un système à se remettre d'une interruption d'infrastructure ou de service, à acquérir dynamiquement des ressources informatiques pour répondre à la demande et à atténuer les perturbations telles que les mauvaises configurations ou les problèmes de réseau transitoires.

Domaines prioritaires

- Principes de conception
- Fondations
- Architecture de charge de travail
- Gestion des modifications
- Gestion des défaillances
- Ressources

Principes de conception

Outre les principes de conception décrits dans le livre blanc AWS Well-Architected Framework, les principes de conception suivants peuvent améliorer la fiabilité des charges de travail liées aux jeux dans le cloud :

- Établissez la base de référence pour le pic de simultanéité des joueurs et les objectifs d'évolutivité du système nécessaires pour atteindre les prévisions commerciales : avant le lancement d'un jeu et pendant les opérations de jeu en direct, établissez des estimations du nombre de joueurs simultanés attendus au pic afin de définir des objectifs cibles d'évolutivité du système afin de répondre à ces projections. Cela permet de créer une base de référence pour la fiabilité de votre jeu. Définissez des politiques de dimensionnement pour s'adapter automatiquement à l'évolution de la demande sans impact sur la disponibilité en vérifiant que vos systèmes de dimensionnement gèrent correctement les sessions actives des joueurs.
- Mesurez votre fiabilité et son impact sur l'expérience des joueurs : définissez des indicateurs de performance clés (KPIs) qui représentent l'état de santé de votre jeu. Surveillez l'impact des modifications apportées à l'infrastructure et aux fonctionnalités du jeu sur votre fiabilité.

Fondations

Pour être fiable, un système doit avoir une base bien planifiée et une surveillance en place avec des mécanismes permettant de gérer les changements de demande ou d'exigences. Le système doit être conçu pour détecter les défaillances et se réparer automatiquement.

Il n'existe pas de bonnes pratiques de base spécifiques au Games Lens. Consultez le livre blanc Well-Architected Framework pour connaître les meilleures pratiques relatives aux bases de la fiabilité applicables aux charges de travail des jeux.

Architecture de charge de travail

GAMEREL01 : Votre architecture de jeu tire-t-elle parti de la résilience du cloud ?

AWS l'infrastructure est construite autour des régions et des zones de disponibilité. Régions AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées à l'aide d'un réseau à faible latence, à haut débit et hautement redondant. Ces structures peuvent être utilisées pour concevoir des charges de travail axées sur des objectifs de fiabilité.

Bonnes pratiques

- GAMEREL01-BP01 Répartissez l'infrastructure de jeu sur plusieurs zones de disponibilité et régions pour améliorer la résilience

GAMEREL01-BP01 Répartissez l'infrastructure de jeu sur plusieurs zones de disponibilité et régions pour améliorer la résilience

Pour minimiser l'impact des défaillances localisées de l'infrastructure sur vos joueurs, vous devez répartir le déploiement de votre infrastructure de manière uniforme sur suffisamment de sites indépendants pour être en mesure de résister à des défaillances inattendues tout en disposant d'une capacité suffisante pour répondre aux besoins de vos joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Lorsque vous déployez votre infrastructure de jeu, il est recommandé de répartir uniformément votre capacité entre plusieurs zones de disponibilité d'une région afin de pouvoir résister aux perturbations d'une ou de plusieurs zones de disponibilité sans perturber l'expérience des joueurs. Les services de backend de jeu tels que les applications Web doivent être équilibrés entre plusieurs zones de disponibilité ou doivent être créés à l'aide d'un service géré tel qu' AWS Lambda Amazon API Gateway, qui assure une haute disponibilité régionale dès la conception. De même, les composants qui maintiennent l'état, tels que les caches, les bases de données, les files d'attente de messages et les solutions de stockage, doivent être conçus pour assurer la persistance durable des données dans plusieurs zones de disponibilité, ce qui est prévu dès la conception dans des services tels qu'Amazon S3, DynamoDB et Amazon SQS, et peut être configuré dans d'autres services.

Lorsque vous concevez l'architecture d'hébergement de votre serveur de jeu dans un souci de résilience, déployez vos flottes de serveurs de jeu de manière uniforme dans les zones de disponibilité au sein de et Région AWS afin de maximiser votre accès à la capacité de calcul disponible dans la région et de réduire l'impact des déficiences des zones de disponibilité. Par exemple, vous pouvez configurer Amazon EC2 Auto Scaling pour utiliser les zones de disponibilité. Si une EC2 instance devient défectueuse, EC2 Auto Scaling peut la remplacer, ainsi que lancer des instances dans d'autres zones de disponibilité si une ou plusieurs zones de disponibilité deviennent indisponibles.

Pour les infrastructures critiques, telles que l'authentification, fournissez un nombre minimum d'instances viables réparties sur plusieurs zones de disponibilité et utilisez le dimensionnement automatique pour gérer les augmentations de charge ou la tolérance aux pannes en cas de défaillance de l'une des zones de disponibilité.

Déployez votre infrastructure de jeu dans plusieurs régions pour optimiser la disponibilité. Les fonctionnalités de reprise après sinistre interrégionales, telles que les bases de données mondiales Aurora et l'infrastructure redondante qui peut devenir active à la suite d'un simple changement de DNS déployé dans une région secondaire, peuvent assurer la continuité du service en cas de défaillance de la région principale. Bien que nous vous encouragions à procéder ainsi afin de garantir la haute disponibilité de vos services de backend de jeu, cette recommandation est particulièrement importante pour vos serveurs de jeux.

Par exemple, dans un jeu multijoueur, la capacité de votre infrastructure pour les serveurs de jeu est susceptible de dépasser les besoins en capacité pour vos autres services, étant donné que les serveurs de jeu sont utilisés pour héberger des sessions de jeu pour les joueurs. De nombreux

jeux choisissent de répartir les joueurs dans des régions de jeu logiques (comme l'ouest et l'est des États-Unis). Pour simplifier l'expérience des joueurs et faciliter l'utilisation d'une infrastructure mondiale pour héberger des jeux, pensez à dissocier le nom de Regions de votre fournisseur de cloud sous-jacent, de la région ou de l'emplacement du centre de données hébergeant physiquement les serveurs de jeu, ainsi que d'autres infrastructures telles que les zones locales ou vos propres centres de données hébergeant des instances de serveur de jeu compatibles avec cette région de jeu.

Lors de la conception de votre service de jumelage, déployez une architecture multirégionale avec des déploiements logiciels distincts entre les régions. Dissociez le déploiement de votre service de matchmaking des flottes qui hébergent vos instances de serveur de jeu afin de pouvoir rediriger les joueurs vers un serveur de jeu dans les régions, quel que soit le déploiement régional de votre service de matchmaking qui a traité la demande de matchmaking.

Concevez une logique dans votre implémentation de matchmaking pour privilégier les régions du serveur de jeu qui répondent à vos règles de latence et à d'autres règles, avec la possibilité de rediriger les joueurs vers d'autres régions si la capacité de vos flottes est faible ou en cas de perturbations de l'infrastructure régionale.

Étapes d’implémentation

- Répartissez l'infrastructure de jeu de manière uniforme sur plusieurs zones de disponibilité pour garantir une disponibilité et une résilience élevées.
- Déployez des services de backend de jeu et des composants dynamiques à l'aide de solutions gérées telles qu'Amazon S3 AWS Lambda, DynamoDB et SQS, ou configurez l'équilibrage de charge et la durabilité pour des solutions personnalisées.
- Mettez en œuvre des déploiements multirégionaux pour les services de jeu et les serveurs critiques, à l'aide de solutions de reprise après sinistre telles que les bases de données mondiales Aurora et les régions logiques orientées vers les joueurs, découplées des emplacements physiques sous-jacents.

Ressources

- Stabilité statique
- Bonnes pratiques concernant les files d'attente pour les sessions de GameLift jeu Amazon
- Flottes Amazon GameLift multirégionales
- Base de données mondiale Aurora pour la reprise après sinistre

Gestion des modifications

GAMEREL02 : Comment adaptez-vous vos jeux dynamiques à l'évolution de la demande ?

Au fur et à mesure que la demande de vos joueurs fluctue au fil du temps, votre infrastructure de jeu doit être capable d'évoluer de manière adaptative pour répondre à ces exigences changeantes. Bien qu'il soit difficile de prévoir la popularité d'un jeu à l'avance, concevez une approche architecturale qui permet d'ajouter ou de supprimer des capacités d'infrastructure pour faire face aux fluctuations de la population de joueurs.

Bonnes pratiques

- GAMEREL02-BP01 Mettre en œuvre une stratégie de mise à l'échelle qui intègre l'état des sessions de jeu actives des joueurs
- GAMEREL02-BP02 Support de l'utilisation de EC2 plusieurs types d'instances pour votre jeu

GAMEREL02-BP01 Mettre en œuvre une stratégie de mise à l'échelle qui intègre l'état des sessions de jeu actives des joueurs

Implémentez une solution permettant de dimensionner automatiquement votre infrastructure de jeu de manière à intégrer la nature dynamique de vos sessions de jeu activement connectées et à gérer avec élégance les activités de mise à l'échelle sans perturber le jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

L'un des avantages du développement d'un jeu dans le cloud est l'élasticité qui peut être obtenue en adaptant automatiquement l'infrastructure des serveurs en fonction des besoins pour répondre à la demande. Alors que les jeux et les services de backend apatrides ou asynchrones peuvent être dimensionnés de manière dynamique à l'aide des politiques d'Amazon EC2 Auto Scaling, de la mise à l'échelle automatique d'EKS ou de techniques similaires généralement adoptées pour les applications Web évolutives, les développeurs de jeux ont généralement besoin d'une approche plus personnalisée pour dimensionner les jeux synchrones ou dynamiques afin de bloquer les interruptions des sessions actives des joueurs.

Étapes d’implémentation

- Pour les jeux dynamiques, générez des statistiques personnalisées qui peuvent être utilisées pour surveiller l'état des sessions de vos joueurs et la capacité disponible du serveur de jeu, qui peuvent être signalées à Amazon CloudWatch sous forme de statistiques personnalisées. Fonctionnalités d'exercice associées à la surveillance des applications, telles que CloudWatch Synthetics, qui vérifient le jeu pour détecter toute altération des fonctions que la simple surveillance ascendante et descendante de l'état de santé risque de ne pas détecter.
- À l'aide de métriques personnalisées, implémentez un logiciel de dimensionnement des serveurs de jeu, par exemple sous forme d'application sans serveur utilisant des AWS Lambda fonctions ou AWS Fargate pour gérer le parc d'instances de serveurs de jeu dédiés en utilisant le AWS SDK pour effectuer des appels d'API afin de mettre à jour les paramètres de capacité minimale, maximale et souhaitée pour les groupes EC2 Auto Scaling hébergeant le build de votre serveur de jeu.
- Utilisez Amazon GameLift pour héberger vos serveurs de jeu et utilisez les fonctionnalités de mise à l'échelle automatique des serveurs de out-of-the-box jeux pour gérer ce processus de dimensionnement à votre place.

Les fonctionnalités de dimensionnement automatique d'Amazon GameLift prennent en compte les sessions de joueurs actives et peuvent être configurées pour bloquer la fermeture ou l'extension des instances de serveur de jeu hébergeant activement des joueurs. Pour plus d'informations, consultez Surveiller GameLift les serveurs Amazon avec Amazon CloudWatch.

GAMEREL02-BP02 Support de l'utilisation de EC2 plusieurs types d'instances pour votre jeu

Lorsque vous hébergez votre jeu à l'aide d' EC2 instances, ou si vous utilisez des conteneurs hébergés sur des EC2 instances de votre Compte AWS entreprise, utilisez plusieurs types d'instances dans votre stratégie d'hébergement. En utilisant plusieurs types d'instances, vous pouvez augmenter le nombre d'options de calcul qui peuvent être utilisées lorsque votre jeu évolue pour ajouter de nouveaux serveurs afin de soutenir la croissance du nombre de joueurs, ce qui améliore la fiabilité au cas où votre type d'instance préféré ne serait pas disponible.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Lorsque vous hébergez votre jeu à l'aide d'instances Spot, utilisez plusieurs types d'instances, car la disponibilité des instances Spot varie en fonction de la demande des clients.

Testez votre jeu sur plusieurs types d'instances pour répondre à vos exigences en matière de coûts et de performances et déterminez un classement par ordre de priorité des types d'instances. Amazon EC2 Auto Scaling prend en charge l'utilisation de plusieurs types et tailles d'instances ainsi que l'attribution de poids à chaque type d'instance dans votre configuration afin que vous puissiez implémenter un classement hiérarchisé des options de calcul.

Lorsque vous hébergez votre jeu à l'aide de l'hébergement GameLift géré par Amazon, déterminez le type d'instances dont votre jeu a besoin et comment exécuter les processus du serveur de jeu sur celles-ci (en utilisant une configuration d'exécution). Lorsque vous choisissez des ressources pour une flotte, tenez compte de plusieurs facteurs, notamment le système d'exploitation du jeu, le type d'instance (le matériel informatique) et l'opportunité d'utiliser des instances à la demande, des instances ponctuelles ou les deux. Les coûts d'hébergement chez Amazon dépendent GameLift principalement du type d'instance que vous utilisez. Pour plus d'informations, voir Choisir des ressources de calcul pour un parc géré.

Étapes d’implémentation

- Utilisez plusieurs types d' EC2 instances pour améliorer la fiabilité et les options de dimensionnement lors de l'hébergement de votre jeu sur EC2 ou dans des conteneurs.
- Configurez Amazon EC2 Auto Scaling ou GameLift des flottes avec des types d'instances et des pondérations prioritaires afin d'optimiser les coûts et les performances.
- Testez votre jeu sur différents types d'instances pour vérifier que les performances répondent aux exigences et ajustez votre stratégie d'hébergement en conséquence .

Gestion des défaillances

GAMEREL03 : Comment maintenez-vous l'état du jeu en cas de perturbation de l'infrastructure ?

Votre infrastructure de jeu étant confrontée à divers événements opérationnels au fil du temps, l'architecture de votre jeu doit être conçue de manière à garantir la continuité de l'expérience

des joueurs et à préserver l'état du jeu lors d'événements liés à l'infrastructure. Pour gérer ces événements, mettez en place des mécanismes de surveillance, d'arrêt progressif et de persistance des états afin de garantir une expérience de jeu fluide pour vos joueurs.

Bonnes pratiques

- GAMEREL03-BP01 Surveillez les perturbations des serveurs de jeu et utilisez les données pour améliorer l'architecture d'hébergement afin d'atteindre les objectifs de fiabilité
- GAMEREL03-BP02 Implémenter un couplage souple des fonctionnalités du jeu pour gérer les échecs avec un impact minimal sur l'expérience des joueurs
- GAMEREL03-BP03 Surveillez les événements de l'infrastructure au fil du temps pour mesurer leur impact sur le comportement des joueurs

GAMEREL03-BP01 Surveillez les perturbations des serveurs de jeu et utilisez les données pour améliorer l'architecture d'hébergement afin d'atteindre les objectifs de fiabilité

Surveillez les indicateurs du serveur de jeu et l'impact des défaillances ou des dégradations de performances, telles que l'augmentation de la latence sous charge, sur le comportement des joueurs au fil du temps afin de pouvoir ajuster la stratégie d'hébergement de votre serveur de jeu pour répondre aux exigences de fiabilité de votre jeu. L'infrastructure du serveur de jeu devant être dégradée doit être immédiatement mise hors service si elle a un impact sur les joueurs ou remplacée de manière proactive lorsqu'aucune session de joueur active n'est hébergée sur le serveur.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Dans les scénarios où les jeux sont hébergés en tant que REST APIs, la fiabilité du système peut être gérée comme les architectures d'applications Web traditionnelles, dans lesquelles le trafic peut être équilibré sur plusieurs serveurs de manière distribuée afin de réduire le risque de défaillance des serveurs.

Pour un jeu synchrone en temps réel, une session de jeu est généralement hébergée sur un processus de serveur de jeu exécuté sur une machine virtuelle, ou une instance de serveur de jeu, car l'état du jeu doit être maintenu de manière performante et répliqué sur les clients de jeu connectés. Cette implémentation signifie que l'expérience d'un joueur est étroitement liée aux performances et à la fiabilité du processus du serveur de jeu qui héberge sa session de jeu. Ce type

d'architecture rend la gestion de la fiabilité des serveurs de jeux plus complexe que les approches traditionnelles.

Pour atténuer l'impact d'une panne de serveur de jeu, configurez votre jeu pour qu'il effectue en permanence des mises à jour asynchrones de l'état de jeu d'un joueur vers un cache ou une base de données hautement disponible telle qu'Amazon ElastiCache (Redis OSS) ou Amazon MemoryDB. En cas de panne du serveur, le dernier état de jeu enregistré du joueur peut être extrait du magasin de données externe et sa session peut être restaurée sur une nouvelle instance de serveur de jeu.

Cependant, cette approche augmente les coûts et la complexité liés à la gestion de cet état externe, et peut ne pas être adaptée aux jeux rapides ou compétitifs où les changements d'état sont si fréquents et se produisent à une telle échelle que l'introduction d'un stockage de données de cache en mémoire, même performant, entraînerait un retard de réplication trop important pour être utile pour restaurer une session. Pour les jeux de cette nature, l'approche optimale consiste à accepter la perte du serveur et à renvoyer le joueur dans un lobby de jeu pour trouver une autre session ou vous pouvez le rediriger automatiquement vers une autre session de jeu.

Capturez autant de données de journal utiles sur la cause de l'interruption du serveur afin de pouvoir étudier le problème ultérieurement. Amazon GameLift fournit des conseils pour résoudre les problèmes de flotte et permet d'accéder à distance aux instances de GameLift flotte Amazon.

Étapes d’implémentation

- Surveillez les indicateurs des serveurs de jeu pour détecter toute dégradation des performances, et supprimez ou remplacez les serveurs dégradés si nécessaire pour garantir la fiabilité.
- Utilisez Amazon ElastiCache ou MemoryDB pour les mises à jour asynchrones de l'état du jeu afin de permettre la restauration de session après une panne de serveur lorsque cela est possible.
- Capturez des données de journal détaillées sur les interruptions des serveurs à des fins d'investigation et de débogage, en tirant parti d'outils tels qu'Amazon GameLift pour la surveillance du parc et l'accès à distance.

GAMEREL03-BP02 Implémenter un couplage souple des fonctionnalités du jeu pour gérer les échecs avec un impact minimal sur l'expérience des joueurs

Le découplage des composants fait référence au concept qui consiste à concevoir des composants de serveur de manière à ce qu'ils puissent fonctionner de manière aussi indépendante que possible.

Certains aspects du jeu sont difficiles à dissocier, car les données doivent être aussi à jour que possible pour offrir une bonne expérience de jeu aux joueurs. Cependant, de nombreux composants et tâches de jeu peuvent être découplés. Par exemple, les classements et les services de statistiques ne sont pas essentiels à l'expérience de jeu, et les lectures et écritures de ces services peuvent être effectuées de manière asynchrone depuis le jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Mettez en œuvre une dégradation progressive des fonctionnalités de votre jeu qui peuvent être désactivées automatiquement ou par un administrateur en cas de détection de problèmes, et configurez les services en amont qui dépendent de la fonctionnalité afin de pouvoir gérer correctement la panne. Par exemple, si des données spécifiques à un joueur ne se chargent pas correctement dans votre client de jeu, vous devez déterminer si ces données sont essentielles à l'expérience de jeu. Si ce n'est pas le cas, configurez le client du jeu pour qu'il gère correctement cet échec sans perturber l'expérience du joueur, en choisissant de réessayer de récupérer ces données ultérieurement lorsque le joueur reviendra sur l'écran.

Utilisez une logique telle que les délais d'expiration, les nouvelles tentatives et les interruptions pour gérer les erreurs et les échecs. Les délais d'attente empêchent les systèmes de se bloquer pendant des périodes déraisonnablement longues. Les nouvelles tentatives peuvent fournir une haute disponibilité des erreurs transitoires et aléatoires.

Définissez les composants non critiques qui peuvent être couplés librement aux composants critiques. Un couplage lâche permet aux systèmes d'être plus résilients, car la défaillance d'un composant ne se répercute pas sur les autres. Lorsque les fonctionnalités du jeu ne nécessitent pas de connexions dynamiques à vos serveurs de jeu ou à votre backend, vous devez implémenter des protocoles sans état pour évoluer de manière dynamique et vous remettre en cas de défaillance passagère. Développez vos composants non critiques là où ils peuvent être librement couplés à des protocoles sans état à l'aide d'une API. HTTP/JSON Mettez en place des appels réseau asynchrones et non bloquants depuis le client du jeu afin de minimiser l'impact sur les joueurs des fonctionnalités de jeu peu performantes ou d'autres services dépendants.

Pour améliorer encore la résilience grâce à un couplage souple, utilisez un service de messagerie tel qu'un système de mise en file d'attente, de streaming ou un système basé sur des sujets entre des composants pouvant être gérés de manière asynchrone. Ce modèle convient aux interactions qui ne nécessitent pas de réponse immédiate ou lorsqu'un accusé de réception indiquant qu'une demande

a été enregistrée est suffisant. Cette solution implique un composant qui génère des événements et un autre qui les consomme. Les deux composants ne s'intégreront pas par point-to-point interaction directe mais par le biais d'un intermédiaire tel qu'une couche de stockage durable ou de mise en file d'attente. Cela contribue également à améliorer la fiabilité du système en préservant les messages en cas d'échec du traitement.

Recherchez et sélectionnez un mécanisme de messagerie approprié, car les différents services de messagerie présentent des caractéristiques différentes, telles que les mécanismes de commande et de livraison. Concevez les opérations de manière à ce qu'elles soient idempotentes afin que le système de messagerie choisi délivre les messages au moins une fois. Par exemple, imaginez un cas d'utilisation typique où votre jeu doit suivre le temps de jeu des joueurs, leurs statistiques ou d'autres données pertinentes, ce qui peut entraîner un débit d'écriture élevé en période de pic de simultanéité des joueurs.

Pour implémenter une architecture fiable, déterminez si le cas d'utilisation nécessite une read-after-write cohérence telle que perçue par le joueur. En général, de tels scénarios sont adaptés au traitement asynchrone et peuvent être réalisés en implémentant un modèle de file d'écriture dans lequel les demandes sont ingérées dans une file de messages évolutive et durable telle qu'Amazon SQS et peuvent être insérées dans votre base de données principale par lots à l'aide d'un service client, tel qu'une fonction Lambda. Cette approche est plus fiable que la communication synchrone entre plusieurs composants distribués, notamment le client de jeu du joueur, vos serveurs Web et d'applications principaux et votre système de base de données interne. Cela réduit également les coûts car la base de données principale n'a pas besoin d'être dimensionnée pour répondre au débit d'écriture maximal, car le traitement client à partir de la file d'attente d'écriture peut être utilisé pour ralentir ce taux d'ingestion selon les besoins.

Étapes d’implémentation

- Dissociez les composants non critiques tels que les classements et les services de statistiques des fonctionnalités de jeu essentielles pour permettre des opérations asynchrones et améliorer la résilience.
- Mettez en œuvre une dégradation progressive pour les fonctionnalités non critiques avec une logique de temporisation, de réessai et d'interruption, et vérifiez que le client du jeu gère les échecs sans perturber l'expérience du joueur.
- Utilisez des systèmes de messagerie tels qu'Amazon SQS pour la communication asynchrone entre les composants, permettant ainsi un traitement évolutif, durable et fiable des cas d'utilisation à haut débit.

Ressources

- Créez des charges de travail hautement évolutives et fiables à l'aide d'une architecture de microservices
- Intégration de microservices à l'aide de services AWS sans serveur
- Comprendre la messagerie asynchrone pour les microservices
- Présentation des modèles de développement de jeux évolutifs sur AWS
- Mise en œuvre de la dégradation progressive

GAMEREL03-BP03 Surveillez les événements de l'infrastructure au fil du temps pour mesurer leur impact sur le comportement des joueurs

Surveillez le processus de votre serveur de jeu, les statistiques de l'instance de votre serveur de jeu et les statistiques de l'expérience de jeu afin de déterminer la cause première des problèmes. Outre la surveillance du processeur et de la mémoire, vous pouvez également configurer la surveillance des métriques réseau liées aux limites du réseau des EC2 instances afin de vous avertir de problèmes tels que le dépassement de bande passante ou d'autres problèmes au niveau du réseau pouvant indiquer que les ressources de votre serveur sont sous-provisionnées. packets-per-second

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Utilisez CloudWatch Synthetics pour vérifier les fonctionnalités critiques de l'application en termes d'expérience du joueur, comme l'impossibilité de se connecter ou d'autres problèmes ayant un impact sur le service. Pour les serveurs de jeux hébergés via Amazon GameLift, pensez à surveiller des indicateurs tels que :

- GameServerInterruptions et InstanceInterruptions qui peut vous aider à comprendre l'impact des limites de disponibilité des instances Spot sur vos serveurs de jeu déployés à l'aide de Spot.
- ServerProcessAbnormalTerminations, qui peut être utilisé pour détecter les interruptions anormales des processus de votre serveur de jeu.

Il est recommandé de conserver les données historiques relatives à la fiabilité de votre serveur de jeu. Utilisez ces données historiques à des fins de reporting et associez-les à d'autres ensembles de données pour découvrir les tendances potentielles et évaluer l'impact sur le comportement des joueurs au fil du temps qui pourrait être dû à des problèmes de serveur de jeu.

Amazon CloudWatch ne conserve pas les métriques indéfiniment, et la résolution de stockage des métriques augmente au fil du temps. Pensez donc à exporter ces métriques vers un stockage à long terme rentable tel qu'Amazon S3. Vous pouvez configurer CloudWatchMetric Streams pour transmettre automatiquement vos métriques depuis CloudWatchRegions vers votre propre compartiment S3, où elles peuvent être stockées à long terme dans un niveau de stockage tel que S3 Intelligent-Tiering, puis archivées à l'aide d'Amazon Glacier. En plaçant vos métriques dans Amazon S3, vous pouvez facilement les associer à d'autres ensembles de données de votre lac de données pour des requêtes interactives avec Amazon Athena.

Étapes d’implémentation

- Surveillez les indicateurs du serveur de jeu, des instances et du réseau, y compris la bande passante et packet-per-second les limites, à l'aide d'Amazon CloudWatch et de CloudWatch Synthetics pour vérifier les fonctionnalités des chemins critiques.
- Suivez GameLift des indicateurs spécifiques tels que GameServerInterruptionset ServerProcessAbnormalTerminationspour évaluer l'impact de la disponibilité des instances Spot et détecter les interruptions anormales des serveurs.
- Exportez CloudWatch des métriques vers Amazon S3 pour un stockage à long terme, utilisez des niveaux économiques tels que S3 Intelligent-Tiering ou Glacier, et analysez les tendances à l'aide d'outils tels qu'Amazon Athena.

Ressources

- Les indicateurs de performance EC2 réseau au niveau des instances Amazon révèlent de nouvelles informations
- CloudWatchStreams métriques : envoyez AWS des métriques à vos partenaires et à vos applications en temps réel

Ressources

Consultez les ressources suivantes pour en savoir plus sur nos meilleures pratiques en matière de fiabilité.

Documents connexes :

- Pratiquer l'intégration continue et la livraison continue sur AWS
- Mise à l'échelle automatique des files d'attente de tâches asynchrones

- Concevez votre WorkloadService architecture
- Expiration des délais, nouvelles tentatives et interruptions dues à la nervosité
- Well-Architected Framework - Pilier de fiabilité
- Architecture pour une évolutivité fiable
- La bibliothèque d'Amazon Builder
- Messagerie en temps réel à grande échelle pour les jeux multijoueurs
- Présentation des modèles de développement de jeux évolutifs sur AWS
- Exécution de microservices conteneurisés sur AWS
- Hébergement d'applications Web dans le cloud
- Création d'une infrastructure réseau multi-VPC évolutive et sécurisée

Vidéos connexes :

- re:Invent 2020 : Ubisoft : Création d'un jeu multijoueur multiplateforme sur AWS
- re:Invent 2018 : Supercell — Scaling Mobile Games
- re:Invent 2019 : Comment CAPCOM crée des jeux amusants avec des conteneurs, des données et du ML
- re:Invent 2018 : Globaliser les comptes des joueurs chez Riot Games tout en préservant la disponibilité
- re:Invent 2020 : GameLoft - Une migration en profondeur vers un data lake sans interruption

Formations associées :

- Utilisation d'Amazon GameLiftFleet IQ pour les serveurs de jeu
- Hébergement de serveurs de jeux avec Amazon EC2

Efficacité des performances

Le pilier de l'efficacité des performances met l'accent sur l'utilisation efficace des ressources informatiques pour répondre aux exigences et sur le maintien de cette efficacité à mesure que la demande évolue et que les technologies évoluent.

Adoptez une approche axée sur les données pour sélectionner une architecture haute performance. Collectez des données complètes sur l'architecture, de la conception de haut niveau à la sélection et à la configuration des types de ressources. Réfléchissez à vos choix architecturaux sur une base cyclique afin de tirer parti d'un ensemble de services et de solutions en constante évolution. Les métriques aident à comprendre les écarts par rapport aux performances attendues afin que vous puissiez prendre des mesures. Une approche axée sur les données permet de faire des compromis avec votre architecture afin d'améliorer les performances, de réduire les coûts ou d'améliorer l'expérience des développeurs.

Domaines d'intérêt

- Principes de conception
- Sélection d’architecture
- Sélection d’une région
- Développement itératif
- Informatique et matériel
- Sélection du calcul
- Gestion des données
- Réseau et diffusion de contenu
- Processus et culture
- Ressources

Principes de conception

Outre les principes de conception décrits dans le livre blanc AWS Well-Architected Framework, les principes de conception suivants peuvent améliorer les performances de vos jeux :

- Mesurez les performances du jeu du point de vue de end-to-end : Il est important de mesurer les performances telles qu'elles sont perçues du point de vue de vos joueurs. Cela signifie que vous devez mesurer les performances du client de jeu, de votre infrastructure de jeu et de la connectivité

Internet qui connecte vos joueurs à l'infrastructure. Cela vous aidera à comprendre les domaines dans lesquels vous pouvez améliorer les performances au sein de votre architecture.

- Optimisez votre architecture pour améliorer les indicateurs qui reflètent l'expérience réelle des joueurs : au fur et à mesure que vous adapterez et ferez évoluer votre architecture, réfléchissez à l'impact de ces améliorations et changements sur l'expérience des joueurs. Les charges de travail des jeux doivent être capables de résister aux défaillances et de minimiser leur impact afin de bloquer les perturbations généralisées du gameplay. Les fonctionnalités de jeu et les systèmes qui ne sont pas étroitement dépendants les uns des autres doivent être découplés afin de réduire le rayon d'impact des pannes et d'isoler les problèmes ayant un impact sur les joueurs.
- Utilisez des technologies qui simplifient le fonctionnement des jeux et accélèrent le développement : privilégiez l'adoption de technologies susceptibles d'améliorer l'efficacité des développeurs. Les frais d'exploitation pendant les phases de développement de pré-production peuvent être une distraction par rapport à l'amélioration du gameplay. En tirant parti des AWS services gérés fournis par AWS nos partenaires, l'ingénierie peut être réduite, ce qui permet aux développeurs de jeux de se concentrer sur le cœur du jeu et sur l'expérience des joueurs. Les exigences en matière d'architecture et de performance peuvent changer et évoluer tout au long du cycle de développement d'un jeu et des compromis technologiques doivent être pris en compte à chaque phase.
- Concevez l'infrastructure pour répondre aux pics de simultanéité des joueurs et évoluez dynamiquement en fonction des besoins : l'infrastructure doit être conçue pour s'adapter à la demande des joueurs. Les indicateurs, tels que la simultanéité des sessions des joueurs et le nombre de connexions, peuvent être utilisés pour effectuer une mise à l'échelle préventive avant que les systèmes ne soient surchargés. Les mesures réactives d'utilisation du système, telles que la consommation du processeur et de la mémoire, peuvent être utilisées pour évoluer en cas de surcharge des systèmes. En adaptant dynamiquement votre infrastructure, vous pouvez réduire les coûts d'exploitation de votre jeu.

Sélection d’architecture

GAMEPERF01 : Comment sélectionnez-vous l'option d'hébergement appropriée pour vos serveurs de jeux ?

La sélection de l'option d'hébergement appropriée pour vos serveurs de jeux est essentielle aux performances de ces serveurs. La décision d'utiliser EC2 des instances, une solution de conteneur

ou un service entièrement géré est l'une des premières décisions à prendre lors de l'architecture pour la production. Chaque option d'hébergement comportera des fonctionnalités et des considérations différentes en matière d'optimisation des performances, de mise à l'échelle, d'exploitation et d'intégration.

Bonnes pratiques

- GAMEPERF01-BP01 Évaluer les besoins en ressources et en évolutivité du serveur de jeu
- GAMEPERF01-BP02 Tenez compte de la surcharge opérationnelle liée à la mise à l'échelle des serveurs de jeu
- GAMEPERF01-BP03 Évaluer l'intégration avec d'autres AWS services, environnements de développement, architectures de processeurs cibles et fonctionnalités

GAMEPERF01-BP01 Évaluer les besoins en ressources et en évolutivité du serveur de jeu

Évaluez les exigences du serveur par rapport à vos besoins d'évolutivité pour vérifier que vous sélectionnez une option d'hébergement qui répond à vos exigences et fournit des performances optimales.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Lorsque vous sélectionnez l'option d'hébergement appropriée pour vos serveurs de jeu, tenez compte des facteurs suivants :

Besoins en ressources du serveur de jeu

Évaluez les exigences en termes de processeur, de mémoire, de réseau et de stockage des processus de votre serveur de jeu afin de déterminer ce que votre jeu consomme. Ne négligez pas le réseau ; chaque image nécessite des cycles de processeur pour recevoir les actions du joueur, mettre à jour l'état du jeu et le renvoyer au joueur. Le déchargement du traitement des paquets peut libérer du processeur pour les fonctions principales du jeu. La mise en réseau est la base d'un jeu fluide et réactif. C'est pourquoi le tester au début du processus permet de définir un profil de performance de référence pour un jeu.

Un jeu de tir à la première personne peut avoir un nombre élevé d'actions par seconde, ce dont le processeur a besoin pour passer rapidement au réseau, ce qui peut favoriser les instances de la

famille C optimisées pour le calcul, tandis qu'un jeu de stratégie au tour par tour qui nécessite plus de cycles de traitement par tour peut nécessiter une augmentation de la mémoire des instances de la famille R pour stocker et mettre à jour l'état du jeu sur le serveur avant de le renvoyer aux joueurs. Utilisez une approche axée sur les données telle que la méthode USE (USE) pour faire des choix architecturaux éclairés.

Évolutivité et élasticité

Évaluez la rapidité et la fluidité avec lesquelles chaque option d'hébergement peut évoluer pour répondre à la demande des joueurs sans compromettre les performances. Tenez compte du niveau d'automatisation et de flexibilité requis pour la charge de travail de votre jeu afin de garantir une expérience de jeu fluide aux heures de pointe. Un serveur de jeu peut évoluer rapidement en augmentant son utilisation en ajoutant des processus de serveur de jeu supplémentaires sur la même instance, tandis qu'un backend de jeu peut évoluer plus lentement en fonction de l'augmentation du nombre d'utilisateurs actifs et des parties jouées. Votre flotte doit évoluer en fonction de la demande afin de minimiser les coûts tout en minimisant les temps d'attente pour que les joueurs entrent dans le jeu. Consultez Amazon EC2 Spot Instance Advisor pour avoir un aperçu de la capacité disponible rentable pour les flottes de serveurs de jeux.

Étapes d’implémentation

- Évaluez les besoins en ressources du serveur de jeu en termes de processeur, de mémoire, de réseau et de stockage afin de sélectionner les types d'instances appropriés, en tenant compte des besoins de performances spécifiques au jeu, tels que le débit réseau élevé pour les jeux FPS ou l'optimisation de la mémoire pour les jeux de stratégie au tour par tour.
- Comparez les différentes options d'hébergement telles que les conteneurs, les instances, le bare-metal et les services gérés en analysant les données de performance à l'aide de frameworks tels que la méthode USE. Utilisez ces informations pour prendre de meilleures décisions concernant l'architecture de votre système.
- Concevez des flottes dans un souci d'évolutivité et d'élasticité, en tirant parti d'outils tels que EC2 Spot Instance Advisor pour optimiser les coûts tout en facilitant une mise à l'échelle rapide afin de répondre à la demande des joueurs pendant les périodes de pointe.

GAMEPERF01-BP02 Tenez compte de la surcharge opérationnelle liée à la mise à l'échelle des serveurs de jeu

Tenez compte des frais de gestion et d'exploitation associés à chaque option d'hébergement.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Frais généraux d'exploitation

Les solutions auto-hébergées sur EC2 ou sur des conteneurs peuvent fournir un meilleur contrôle, mais nécessitent également une gestion plus poussée. Les orchestrateurs de conteneurs tels qu'ECS ou EKS peuvent réduire les temps de lancement des serveurs conteneurisés tout en augmentant la complexité du réseau et les frais d'orchestration de maintenance.

À titre d'exemple, les groupes de nœuds gérés par EKS peuvent automatiser le provisionnement et la gestion du cycle de vie de vos serveurs de jeu, mais ils ne respectent pas les budgets relatifs aux interruptions de service lors de la fermeture d'un nœud. Si votre jeu nécessite plus de 15 minutes pour terminer les parties en toute sécurité, vous devrez peut-être créer des hooks de cycle de vie ou envisager des nœuds autogérés dotés de manettes personnalisées pour bloquer les interruptions de jeu.

Les services gérés tels qu'Amazon Game Lift peuvent prendre en charge la majeure partie des frais opérationnels, mais ils réduisent la visibilité et le contrôle des exigences particulières relatives à la configuration réseau et de sécurité de bas niveau. Le choix d'une solution de serveur de jeu est un compromis entre le niveau de personnalisation, de contrôle et de responsabilité que vous aurez à assumer pour optimiser les performances du serveur de jeu et le dimensionnement du comportement.

Étapes d’implémentation

- Évaluez les coûts opérationnels liés aux options d'hébergement, en équilibrant les efforts de contrôle et de gestion entre les solutions auto-hébergées telles que EC2 ECS ou EKS et les services gérés tels qu'Amazon Game Lift.
- Utilisez les groupes de nœuds gérés par EKS pour l'automatisation, mais implémentez des hooks de cycle de vie ou des contrôleurs personnalisés si vos serveurs de jeu nécessitent des périodes d'interruption plus longues que celles par défaut.
- Évaluez les compromis entre personnalisation, visibilité et responsabilité opérationnelle lors de la sélection d'une solution de serveur de jeu.

GAMEPERF01-BP03 Évaluer l'intégration avec d'autres AWS services, environnements de développement, architectures de processeurs cibles et fonctionnalités

Évaluez dans quelle mesure chaque option d'hébergement s'intègre aux autres AWS services sur lesquels repose votre jeu, tels que les bases de données, les analyses ou les services de diffusion de contenu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Vous pouvez exploiter facilement d'autres services AWS

L'intégration fluide entre les services offre des avantages opérationnels tels qu'une meilleure surveillance des performances et une transmission sécurisée efficace des données entre les composants de jeu, les serveurs de jeux, les services de backend de jeu et les solutions d'observabilité.

Par exemple, la coordination des changements de trafic pour les jeux en direct peut s'avérer complexe. Amazon Route 53 vous aidera à maintenir vos enregistrements DNS à jour, ce qui simplifie les transferts de trafic coordonnés. AWS Les numéros de trafic de Global Accelerator vous permettent d'envoyer un pourcentage du trafic vers une autre région et d'assurer le bon fonctionnement de votre jeu pendant la maintenance.

Environnement et outils de développement

Tenez compte des outils de développement, des frameworks et des environnements pris en charge par chaque option d'architecture. Vérifiez que l'option que vous avez choisie correspond à votre solution de développement de jeux et à vos langages de programmation, car cela peut avoir un impact sur la capacité de votre équipe à optimiser et à maintenir les performances du serveur de jeu. La diffusion d'un jeu sur mobile, console et PC augmentera la complexité des outils et des tests. Le support multisystème est particulièrement important pour les studios multijeux où les services centralisés peuvent standardiser les meilleures pratiques de développement pour tous les titres.

Architecture et fonctionnalités du processeur cible

Tenez compte du profil de performance de votre moteur de jeu et des processus de votre serveur de jeu, ainsi que du niveau de support ARM disponible. Évaluez si vous pouvez bénéficier de l'amélioration du rapport prix/performances des AMD64 processeurs Graviton basés sur ARM ou x86.

Devez-vous utiliser des fonctionnalités Intel telles que le chiffrement AES-NI, AVX ou Turbo Boost ? Passez en revue les types d'hôtes dédiés pour identifier les familles d'instances à un ou plusieurs sockets. Lorsque vous utilisez une famille d'instances multisocket, pensez à épingler NUMA et à partager le cache L3 dans les processus de votre serveur de jeu. Utilisez les configurations C-state et P-state pour optimiser les performances de votre jeu en réglant l'horloge de fréquence et en réduisant le niveau de sommeil.

Étapes d’implémentation

- Sélectionnez des options d'hébergement offrant une intégration parfaite avec AWS des services tels AWS Secrets Manager qu'ACM et d'autres, afin de rationaliser le suivi des performances, de sécuriser la livraison des données et de réduire les tâches opérationnelles manuelles.
- Vérifiez la compatibilité entre votre option d'hébergement et votre environnement de développement, vos frameworks et vos langages de programmation afin d'optimiser et de maintenir efficacement les performances du serveur.
- Évaluez les exigences en matière d'architecture du processeur, en tirant parti de Graviton pour le rapport prix/performances ou de x86 pour des fonctionnalités spécifiques telles que AES-NI, AVX et Turbo Boost, et optimisez les performances du serveur grâce à l'épinglage NUMA et au réglage C-State/P-State.

Sélection d’une région

GAMEPERF02 : Comment déterminez-vous les régions géographiques qui hébergeront votre infrastructure de jeu ?

Le choix de l'emplacement idéal pour votre infrastructure de jeu peut améliorer les performances réseau pour les joueurs et le backend. Il est important de tenir compte de l'origine des connexions de votre base de joueurs et de la manière dont vos communautés ou vos serveurs sont construits pour assurer la croissance et la durabilité à long terme d'une région géographique. Le déploiement d'une infrastructure de serveur de jeu découplée et de services principaux peut améliorer votre efficacité opérationnelle globale et votre résilience en utilisant plusieurs régions, zones locales et avant-postes pour héberger votre jeu.

Bonnes pratiques

- GAMEPERF02-BP01 Sélectionnez une région proche de vos joueurs

- GAMEPERF02-BP02 Concevez une approche qui permet de placer une infrastructure de jeu sensible à la latence à proximité des joueurs afin d'améliorer les performances

GAMEPERF02-BP01 Sélectionnez une région proche de vos joueurs

Lors du lancement initial d'un jeu, vous devez déterminer où déployer l'infrastructure sur la base de discussions avec les parties prenantes de votre entreprise, telles que les équipes de publication qui déterminent où le jeu doit être mis à la disposition des joueurs et où elles concentrent leurs efforts marketing et publicitaires avant le lancement.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les parties prenantes de votre entreprise devraient également disposer de mécanismes pour stimuler la demande afin de mieux comprendre l'accueil et la viabilité des joueurs. Par exemple, ces équipes disposeront de mécanismes tels que des précommandes de jeux, des campagnes et des événements marketing, des listes d'e-mails publiques permettant aux joueurs de manifester leur intérêt avant le lancement, et d'autres approches pour établir des signaux pertinents afin de déterminer où le jeu aura probablement le plus de joueurs au lancement. Le jeu peut également utiliser une stratégie de déploiement régional comprenant des phases de test de jeu et de lancement progressif afin de déterminer la demande des joueurs régionaux.

Sélectionnez une région d'origine proche de votre base de joueurs et de vos développeurs et dotée des AWS services et fonctionnalités dont vous avez besoin pour héberger votre jeu. La maison RSegion sera l'endroit où s'exécuteront les services de backend du jeu, et elle peut également faire fonctionner les serveurs de jeu. Évaluez une région d'origine en fonction des services pris en charge, de la connectivité aux emplacements périphériques, de la proximité des régions de basculement et du nombre de zones de disponibilité. Si vous utilisez une zone locale, considérez que la région parent est parfois située dans une zone géographique différente. Par exemple : Santiago, au Chili, la zone locale us-east-1-scl-1a a pour région mère la Virginie du Nord us-east-1, même si elle est géographiquement plus proche de Sao Paulo sa-east-1.

Étapes d’implémentation

- Identifiez les régions de déploiement en fonction des signaux de demande des joueurs issus des activités de pré-lancement, telles que les précommandes, les campagnes marketing et les inscriptions d'intérêt.

- Choisissez une région d'origine proche de la base de joueurs et des développeurs principaux, en vous assurant qu'elle prend en charge les AWS services requis, les emplacements périphériques et les régions de basculement.
- Évaluez soigneusement les zones locales, en tenant compte du fait que la région mère peut différer géographiquement de l'emplacement de la zone locale.

GAMEPERF02-BP02 Concevez une approche qui permet de placer une infrastructure de jeu sensible à la latence à proximité des joueurs afin d'améliorer les performances

Le placement séparé des infrastructures sensibles à la latence, telles que les serveurs de jeux, minimise l'impact des longs itinéraires réseau. Les déploiements répétables peuvent simplifier la gestion de plusieurs sites plus performants pour vos joueurs. Le ping est une métrique courante qui apparaît dans l'interface utilisateur du jeu et un ping faible peut être une capacité de différenciation.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Lorsque vous lancez un jeu pour la première fois, il se peut que vous ne disposiez pas encore de suffisamment d'informations sur votre base de joueurs pour savoir exactement où déployer l'infrastructure la plus proche des joueurs les plus intéressés par votre jeu. Il s'agit d'un défi courant, et vous devez vous préparer à ce scénario en concevant une architecture qui vous permette d'ajuster rapidement votre stratégie de placement d'hébergement afin de déployer des serveurs là où ils sont nécessaires, plus près des joueurs. Les développeurs de jeux évaluent généralement le déploiement de leur infrastructure de jeu dans le cadre d'une analyse récurrente après le lancement afin d'investir progressivement dans des améliorations au fil du temps grâce à une approche itérative.

Une bonne pratique consiste à utiliser des infrastructure-as-code modèles, tels que AWS CloudFormation Terraform by Hashicorp, pour la configuration de votre infrastructure VPCs, notamment les configurations de sous-réseaux et les dépendances requises pour lancer des services de jeu critiques, afin de pouvoir vous référer à ces modèles, les personnaliser rapidement si nécessaire et les déployer là où une infrastructure supplémentaire est nécessaire pour soutenir vos joueurs.

Vous devez également vous assurer de comprendre comment votre stratégie de déploiement actuelle pourrait être modifiée pour permettre une future expansion. Les modèles IaC sont reproductibles mais ne remplacent pas la planification du réseau. IPAM gère votre. VPCs Dimensionnement du

sous-réseau, sélection de la zone de disponibilité, inventaire des adresses IP et alignement des zones de disponibilité entre comptes. Le réseau est important à prendre en compte et peut perturber les joueurs en cas de modification. Les serveurs de jeu déployés sur plusieurs sites géographiques se connecteront à votre backend de jeu, qui est généralement hébergé dans une ou plusieurs régions d'origine, ce qui peut nécessiter une configuration supplémentaire pour prendre en charge la connectivité privée. Ces considérations doivent être continuellement évaluées au fil du temps afin que vous puissiez modifier votre stratégie d'hébergement de jeux en fonction de l'évolution des exigences de votre jeu ou de celles de vos joueurs.

Lorsque vous déterminez le nombre d'emplacements d'hébergement de jeux à utiliser pour votre jeu, tenez compte des facteurs suivants :

- Amélioration de la qualité de l'expérience des joueurs : dans quelle mesure pouvez-vous améliorer l'expérience des joueurs en ajoutant des sites d'hébergement de jeux supplémentaires ? Quel est le gain de performance progressif que vous pouvez obtenir en procédant ainsi ? Comment allez-vous mesurer cette amélioration des performances ?
- Quelles populations de joueurs prioriser : pour combien de joueurs pouvez-vous améliorer l'expérience si vous ajoutez des sites d'hébergement de jeux supplémentaires ? Quelles populations de joueurs, ou quelles zones géographiques, allez-vous prioriser ?
- Impacts du changement en aval : Si vous modifiez votre stratégie d'hébergement de jeux, comment cela influencera-t-il vos temps d'attente pour le matchmaking pour les joueurs ? La taille des parties, l'équilibre des compétences ou le nombre de joueurs dans le pool de joueurs peuvent-ils s'adapter à un changement de stratégie en matière de lieu d'hébergement du jeu ? La prise en charge d'un plus grand nombre de sites peut potentiellement fragmenter le pool de joueurs et augmenter les coûts et la complexité.

Chacune de ces considérations doit être évaluée lorsque vous déterminez où vous ajoutez ou supprimez des sites d'hébergement de jeux. Par exemple, vous pouvez choisir de donner la priorité à l'amélioration de l'expérience pour les joueurs situés dans des zones géographiques où l'expérience de jeu est la moins performante, ou pour les joueurs qui expriment le plus de commentaires publics. Vous pouvez également choisir de prendre en compte la monétisation des joueurs dans vos priorités, par exemple en vous concentrant sur l'amélioration de l'expérience des joueurs situés dans des zones géographiques qui génèrent une source de revenus significative pour votre jeu ou qui sont susceptibles de générer des revenus supplémentaires si vous améliorez les performances.

Outre l'hébergement de l'infrastructure Régions AWS, vous pouvez utiliser les Zones Locales, qui sont une extension d'un Région AWS, pour héberger vos serveurs de jeu et d'autres applications

sensibles à la latence, telles que des serveurs de chat vocal plus proches de vos joueurs. Vous pouvez également choisir de gérer l'infrastructure de développement de jeux dans les Zones Locales afin d'améliorer l'expérience de vos équipes de développement de jeux. Par exemple, vous pouvez utiliser les zones locales pour traiter des cas d'utilisation tels que l'hébergement de répliques de vos serveurs de contrôle de source autogérés plus près de vos développeurs de jeux, et pour proposer des stations de travail virtuelles de développement de jeux et un stockage de contenu aux utilisateurs utilisant des EC2 instances Amazon, des volumes EBS et des systèmes de FSx fichiers Amazon déployés dans une ou plusieurs zones locales à proximité de vos studios de développement sans que vous ayez à héberger l'infrastructure sur site.

Les Outposts sont un bon choix lorsque les Régions ou les Zones Locales ne sont pas disponibles dans la même zone géographique. La connectivité entre votre centre de données et votre système AWS doit être prise en compte pour garantir la fiabilité du serveur de jeu au système principal. AWS Outposts et les serveurs Outpost sont spécialement conçus pour fonctionner AWS dans votre centre de données à l'aide des mêmes services et pour aider APIs à créer un modèle de déploiement cohérent quel que soit l'endroit où vous exécutez votre jeu. Plusieurs racks peuvent être combinés dans un avant-poste logique, et l'infrastructure peut être partagée entre eux. Comptes AWS Le cycle de vie du matériel est géré par AWS et le délai de livraison peut être aussi court que 3 mois.

Si vous créez des jeux à l'aide de conteneurs et que vous souhaitez pouvoir adopter une architecture de déploiement hybride utilisant un logiciel open source pouvant être déployé sur votre propre infrastructure sur site, vous pouvez utiliser ECS Anywhere ou EKS Anywhere comme alternative aux Zones AWS Outposts Locales. Si vous hébergez sur Amazon GameLift, Amazon GameLift Anywhere peut être utilisé pour faire fonctionner votre serveur sur du matériel local, ce qui peut accélérer votre processus de développement, vous permettre d'utiliser des zones locales ou d'enregistrer votre propre métal dans votre flotte.

Étapes d’implémentation

- Utilisez infrastructure-as-code des outils tels que AWS CloudFormation Terraform pour des déploiements reproductibles, ce qui permet de personnaliser et d'adapter rapidement les sites d'hébergement de jeux en fonction des besoins des joueurs.
- Évaluez les améliorations de l'expérience des joueurs, les priorités relatives à la population de joueurs et les impacts en aval, tels que les temps de matchmaking lors de l'ajout ou de la suppression d'emplacements d'hébergement de jeux.
- Utilisez AWS des Zones Locales, des Outposts ou des options hybrides comme ECS Anywhere, EKS Anywhere ou GameLift Anywhere pour optimiser l'infrastructure sensible à la latence et répondre à divers besoins de déploiement.

Développement itératif

GAMEPERF03 : Comment utiliser Amazon GameLift pour optimiser les performances de développement itératif ?

Amazon GameLift fournit un end-to-end flux de travail pour le développement et les tests de performance de votre jeu dans un environnement de test local.

Bonnes pratiques

- GAMEPERF03-BP01 Utilisez GameLift Amazon Anywhere et une boîte à outils de test GameLift
- GAMEPERF03-BP02 Testez les performances et l'évolutivité des serveurs de jeux
- GAMEPERF03-BP03 Optimiser l'utilisation des ressources des conteneurs GameLift

GAMEPERF03-BP01 Utilisez GameLift Amazon Anywhere et une boîte à outils de test GameLift

Pour améliorer l'efficacité des performances grâce à un processus de développement itératif, utilisez Amazon GameLift Anywhere avec l'Amazon GameLift Testing Toolkit pour créer un environnement de test complet.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Cette approche permet une itération rapide, une collecte de données efficace et une analyse détaillée des performances. Les principales étapes sont les suivantes :

Création d'un environnement de test

Utilisez Amazon GameLift Anywhere pour configurer un environnement de test local ou basé sur le cloud. Cette configuration élimine le besoin de télécharger chaque itération de build du serveur de jeu vers une flotte gérée, réduisant ainsi le temps d'activation.

Intégrer Amazon GameLift Testing Toolkit

Intégrez l'Amazon GameLift Testing Toolkit à votre flux de travail de développement. La boîte à outils fournit des scripts, des outils et des bibliothèques permettant de visualiser GameLift l'infrastructure

Amazon, de lancer des joueurs virtuels et d'itérer des ensembles de FlexMatch règles à l'aide du FlexMatch simulateur. Il simplifie l'intégration et la gestion des GameLift ressources Amazon, vous permettant d'automatiser les tâches courantes et de collecter les données nécessaires à l'analyse des performances.

Cycles de construction et de test rapides

Mettez rapidement à jour le parc de tests avec de nouvelles versions, démarrez-le et commencez les tests. Cela facilite un build-test-repeat cycle rapide, permettant aux développeurs de valider différents aspects de l'expérience du joueur, y compris les interactions multijoueurs.

Tests complets

Testez l'intégration de votre serveur de jeu avec le SDK GameLift du serveur Amazon, les interactions avec les services principaux, les configurations de matchmaking et les autres fonctionnalités d' GameLift hébergement. Utilisez le kit de GameLift test pour automatiser les tests et recueillir des indicateurs de performance détaillés, afin de vous assurer que les composants du jeu fonctionnent parfaitement ensemble.

Analyser les données de performance

Utilisez les données collectées par le GameLift Testing Toolkit pour analyser les problèmes de performance et optimiser votre serveur de jeu. La boîte à outils permet de suivre les indicateurs clés, d'identifier les problèmes et de prendre des décisions basées sur les données afin d'améliorer l'efficacité des performances.

En intégrant Amazon GameLift Anywhere et le GameLift Testing Toolkit à votre processus de développement itératif, vous pouvez améliorer considérablement l'efficacité des performances grâce à des tests rapides, à des contrôles d'intégration complets et à une analyse détaillée des performances.

Étapes d’implémentation

- Utilisez Amazon GameLift Anywhere pour créer un environnement de test, en réduisant le temps d'activation pour les builds de serveurs de jeu et en permettant une itération rapide.
- Intégrez l'Amazon GameLift Testing Toolkit pour automatiser les tâches de test, simuler les joueurs et valider FlexMatch les configurations pendant le développement.
- Collectez et analysez les données de performance avec le kit de GameLift test pour identifier les goulots d'étranglement, optimiser les serveurs de jeu et améliorer l'efficacité des performances.

GAMEPERF03-BP02 Testez les performances et l'évolutivité des serveurs de jeux

Pour tester les performances et l'évolutivité de vos serveurs de jeu, implémentez un framework de test robuste utilisant les GameLift fonctionnalités d'Amazon et le GameLift Testing Toolkit.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les principales pratiques sont les suivantes :

Tests itératifs

Utilisez une flotte Amazon GameLift Anywhere pour créer un environnement hébergé dans le cloud dans lequel vous pouvez créer et tester de manière itérative des composants de jeu. Cet environnement doit refléter les conditions d'hébergement réelles, afin de permettre des tests de performance et d'évolutivité réalistes.

Tests d'intégration des serveurs de jeux

Testez l'intégration de votre serveur de jeu au SDK du GameLift serveur Amazon, notamment en démarrant de nouvelles sessions de jeu et en suivant les événements des sessions de jeu à l'aide AWS CLI du GameLift Testing Toolkit. Cela permet de vérifier que le serveur de jeu fonctionne correctement dans l' GameLift environnement.

Utilisez le kit GameLift d'outils de test pour automatiser les tests et recueillir des indicateurs de performance détaillés. La boîte à outils vous permet de visualiser GameLift l'infrastructure, de lancer des joueurs virtuels pour les tests de charge et d'itérer des ensembles de FlexMatch règles avec le FlexMatch simulateur. Il est particulièrement utile pour dimensionner les tâches ECS Fargate, qui simulent les sessions des joueurs en créant de nombreuses sessions de jeu simultanées afin de tester le stress de l'infrastructure du serveur.

Tests d'évolutivité

Testez des modèles de files d'attente pour les sessions de jeu, des flottes multi-sites, des flottes ponctuelles et à la demande et plusieurs types d'instances. Testez les options de placement des sessions de jeu, les politiques de latence et les paramètres de priorisation de la flotte. Configurez le dimensionnement de la capacité pour répondre à la demande des joueurs et vérifiez que le système peut gérer la charge attendue dans différentes conditions.

Étapes d’implémentation

- Utilisez Amazon GameLift Anywhere pour configurer un environnement de test réaliste pour des tests itératifs de performance et d'évolutivité.
- Testez l'intégration du serveur de jeu avec le SDK GameLift du serveur, afin de faciliter la gestion correcte des sessions et le suivi des événements dans l' GameLift environnement.
- Effectuez des tests d'évolutivité avec le kit de GameLift test, en simulant la charge de joueurs, en testant les files d'attente des sessions et en validant le dimensionnement de la flotte, les politiques de latence et les paramètres de priorisation.

GAMEPERF03-BP03 Optimiser l'utilisation des ressources des conteneurs GameLift

Pour optimiser l'utilisation des ressources des GameLift conteneurs, concevez efficacement votre flotte de conteneurs et définissez des limites de ressources précises.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les principales directives sont les suivantes :

- Conception de groupes de conteneurs : organisez vos logiciels en groupes de conteneurs. Le conteneur principal doit regrouper votre application de serveur de jeu et l' GameLift agent Amazon. Utilisez des conteneurs annexes pour des logiciels supplémentaires afin de gérer les dépendances et de définir des limites spécifiques à chaque conteneur pour l'utilisation de la mémoire et du processeur.
- Définissez des limites de ressources : pour chaque groupe de conteneurs, déterminez les ressources de mémoire et de processeur requises. Définissez des limites facultatives pour les conteneurs individuels afin de vérifier qu'ils disposent de ressources réservées, mais qu'ils peuvent également dépasser ces limites si des ressources supplémentaires sont disponibles. Cela permet d'éviter les conflits de ressources et les défaillances potentielles des conteneurs.
- Groupe de conteneurs de démons : envisagez d'utiliser un groupe de conteneurs de démons pour les processus d'arrière-plan ou de surveillance qui n'ont pas besoin d'évoluer avec le groupe de conteneurs principal. Cela permet de vérifier que les tâches d'arrière-plan essentielles sont gérées efficacement sans affecter les processus du serveur de jeu principal.

Étapes d’implémentation

- Concevez des groupes de conteneurs avec un conteneur principal pour le serveur de jeu et l' GameLift agent, et des sidecars pour gérer les dépendances, avec des limites de mémoire et de processeur spécifiques.
- Définissez des limites de ressources pour chaque groupe de conteneurs afin de réserver les ressources requises tout en permettant une utilisation contrôlée des ressources afin d'éviter les conflits.
- Utilisez un groupe de conteneurs de démons pour les tâches d'arrière-plan ou de surveillance, afin de vous assurer qu'elles fonctionnent efficacement sans affecter les processus du serveur de jeu principal.

Informatique et matériel

GAMEPERF04 : Comment empêcher les sessions de jeu d'avoir un impact sur les joueurs utilisant la même instance de serveur de jeu ?

Une fois votre serveur de jeu lancé AWS, vous devez surveiller ses performances pour offrir une expérience de qualité aux joueurs, indépendamment de l'utilisation des ressources, du calcul sous-jacent ou de la saturation.

Bonnes pratiques

- GAMEPERF04-BP01 Surveiller les processus du serveur de jeu pour détecter les problèmes
- GAMEPERF04-BP02 Testez les performances de votre serveur de jeu avec des scénarios de jeu simulés et réels

GAMEPERF04-BP01 Surveiller les processus du serveur de jeu pour détecter les problèmes

Vous pouvez exécuter plusieurs processus de serveur de jeu par instance pour utiliser efficacement les ressources de vos instances de serveur de jeu. Si tel est le cas, concevez votre architecture de telle sorte qu'un processus de serveur de jeu individuel hébergeant une session de jeu ne puisse pas avoir d'impact négatif sur les autres sessions de jeu hébergées sur la même instance. Utilisez des

indicateurs pour comprendre l'impact du placement et du type de mode de jeu sur les performances des instances de serveur de jeu. Intégrez un mélange de processus à faible charge (lobby, boutique ou tutoriel solo) et à charge élevée (gameplay classé, multijoueur ou exigeant) pour éviter de vous retrouver dans l'instance du serveur de jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Surveillez l'expérience des joueurs à l'aide de statistiques côté client et côté serveur en collectant des données télémétriques relatives au temps de ping et à l'instabilité, aux pertes d'images, au temps de réponse des API, aux erreurs et à la réussite de la boucle de jeu. Corrélez les horodatages de ces événements avec les problèmes d'assistance aux joueurs et les journaux du serveur afin d'identifier les goulots d'étranglement liés aux performances. Des outils tels que Dtrace, ftrace, uperf et eBPF peuvent être utilisés pour une investigation et une analyse approfondies des performances du système.

Mettez en place une surveillance des ressources limitées disponibles pour vos instances de serveur de jeu afin de pouvoir générer des alertes lorsque les processus individuels du serveur de jeu dépassent les seuils de budget de ressources prédéterminés. Lorsque les seuils sont dépassés, vous souhaiterez peut-être configurer le logiciel de votre serveur de jeu pour transférer les déconnexions du système et du serveur de jeu pertinentes vers un stockage durable, tel qu'une solution de journalisation centralisée, afin que les ingénieurs de vos serveurs de jeux puissent étudier ce comportement. En outre, votre instance de serveur de jeu doit être configurée pour rapporter les métriques de chacun des processus de serveur de jeu exécutés sur l'instance afin que vous puissiez surveiller ces processus individuels en plus des mesures globales pour l'instance de serveur de jeu.

Par exemple, GameLift fournit des statistiques pour la surveillance des sessions de jeu, qui peuvent être complétées par des statistiques personnalisées spécifiques au jeu et des journaux collectés à l'aide de l' CloudWatchagent Amazon que vous pouvez configurer sur votre instance de serveur de jeu. Vos statistiques peuvent être consultées CloudWatch ou exportées vers d'autres outils tels qu'Amazon Managed Grafana, qui est intégré à Single Sign-On pour permettre aux utilisateurs qui n'ont peut-être pas accès à la console de gestion d'accéder facilement aux métriques. Reportez-vous aux bonnes pratiques suivantes pour gérer les journaux et les statistiques à l'aide d'Amazon GameLift, qui fournit également une assistance pour consulter les journaux de sessions de jeu individuels.

Étapes d’implémentation

- Exécutez plusieurs processus de serveur de jeu par instance et mélangez les modes de jeu à faible charge et à charge élevée pour éviter les points chauds et garantir une utilisation équilibrée des ressources.
- Surveillez les indicateurs côté client et côté serveur tels que le ping, le jitter, les pertes d'images et les temps de réponse des API, et corrélez-les aux journaux du serveur et aux problèmes signalés par les joueurs afin d'identifier les goulots d'étranglement.
- Configurez la surveillance des ressources pour chaque processus du serveur de jeu, générez des alertes en cas de dépassement des seuils et stockez les journaux dans un stockage durable à des fins d'analyse à l'aide d'outils tels CloudWatch qu'Amazon Managed Grafana.

GAMEPERF04-BP02 Testez les performances de votre serveur de jeu avec des scénarios de jeu simulés et réels

Effectuez des tests de performance et évaluez différents scénarios de jeu pour déterminer si le processus du serveur de jeu gère correctement l'utilisation des ressources fixes, telles que la mémoire d' EC2 instance, le processeur et la bande passante réseau.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

La création de tests de jeu simulés avec des robots capables de refléter les trajectoires de jeu et les comportements courants de vos joueurs peut déterminer la manière dont les processus de votre serveur de jeu gèrent cela selon différents scénarios d'utilisation. Par exemple, vous pouvez implémenter une solution, telle que le test de charge distribué, AWS que vous pouvez personnaliser pour exécuter des simulations de clients de jeu ou des builds de clients de jeu pour générer des scénarios de jeu. Réalisez des tests de jeu internes et faites appel à des équipes d'assurance qualité pour tester le stress des différentes fonctionnalités de votre jeu afin de vous assurer que votre jeu est conçu pour fonctionner de manière optimale. AWS Device Farmpeut être utilisé pour effectuer des tests mobiles et Web pour vos jeux iOS, Android et par navigateur sur plusieurs types d'appareils.

Étapes d’implémentation

- Effectuez des tests de performance avec des robots simulant les comportements courants des joueurs afin d'évaluer l'utilisation des ressources du serveur de jeu selon différents scénarios.

- Utilisez des solutions telles que le test de charge distribué AWS pour personnaliser et simuler des scénarios de jeu pour les tests de stress.
- Réalisez des tests de jeu internes et utilisez des outils tels que AWS Device Farm les tests de jeux mobiles et sur navigateur sur différents appareils.

Sélection du calcul

GAMEPERF05 : Comment choisissez-vous la solution informatique adaptée à votre jeu ?

Les performances de calcul varient en fonction de la taille et de la famille d'instances. Il est avantageux d'utiliser plusieurs options de calcul provenant de pools de capacité distincts. Développez une stratégie de composition de flotte qui privilégie les performances, mais qui inclut suffisamment de diversité pour éviter les erreurs de capacité insuffisante.

Bonnes pratiques

- GAMEPERF05-BP01 Comparez les performances de votre jeu sur plusieurs types de calcul
- GAMEPERF05-BP02 Déplacer les tâches de non-latency-sensitive calcul vers des flux de travail asynchrones

GAMEPERF05-BP01 Comparez les performances de votre jeu sur plusieurs types de calcul

En ce qui concerne les charges de travail des serveurs de jeux, il n'existe pas d'approche unique pour identifier la solution informatique optimale pour héberger votre serveur de jeu. Une stratégie courante pour l'analyse comparative des serveurs de jeux consiste à commencer par des instances EC2 « c » optimisées pour le calcul, car cette famille d'instances fournit des performances élevées pour les charges de travail gourmandes en calculs. Par ailleurs, si votre jeu nécessite une quantité importante de mémoire pour implémenter des fonctionnalités spécifiques, les instances optimisées en mémoire peuvent être les plus appropriées.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Si votre charge de travail utilise d'importantes ressources réseau, pensez à implémenter des instances optimisées pour le réseau, ce qui est généralement indiqué par un « n » dans le nom de l'instance. Évitez les instances de type « t » éclatantes, car une fois les crédits épuisés, les performances diminueront. Les jeux étant sensibles à la latence et aux pertes de paquets, il est recommandé d'utiliser EC2 une mise en réseau améliorée pour améliorer les performances réseau de vos serveurs de jeux. La mise en réseau améliorée utilise la I/O virtualisation à racine unique (SR-IOV) pour fournir des fonctionnalités réseau hautes performances sur les types d'instances pris en charge. Le SR-IOV est une méthode de virtualisation des appareils qui fournit des I/O performances supérieures et une faible utilisation du processeur par rapport aux interfaces réseau virtualisées traditionnelles. La mise en réseau améliorée offre une bande passante supérieure, des performances de paquet par seconde (PPS) nettement plus élevées, ainsi que des latences réduites entre les instances. La mise en réseau améliorée avec Elastic Network Adapter est disponible pour les types d' EC2 instances les plus récents et il est important de procéder à des mises à jour régulières afin de bénéficier des améliorations de performances apportées par les nouvelles instances et des améliorations apportées à l'AWS hyperviseur Nitro.

Si votre jeu fonctionne de la même manière sur plusieurs types d' EC2 instances, vous devriez envisager d'utiliser plusieurs types d'instances pour héberger vos serveurs de jeu. Surveillez les performances au fil du temps et optimisez davantage une fois que vous avez organisé suffisamment de sessions de jeu en production pour être en mesure d'identifier les tendances en matière de performances. N'oubliez pas que vos exigences de calcul peuvent changer à mesure que vous ajoutez de nouvelles fonctionnalités à votre jeu qui nécessitent une allocation de ressources différente. Vous pouvez configurer les groupes EC2 Auto Scaling pour qu'ils utilisent plusieurs types d'instances, ou vous pouvez utiliser des groupes Auto Scaling distincts pour héberger des instances de serveurs de jeu qui exécutent des types d'instances distincts, ce qui peut simplifier la gestion de la corrélation et de l'agrégation des métriques.

Évaluez les performances de votre jeu sur différents types de processeurs tels que les instances Intel, les instances AMD et les instances Graviton basées sur ARM. Unreal Engine 5.1.1 ou version ultérieure peut compiler des serveurs de jeu pour Graviton et améliorer le rapport qualité-prix de votre jeu. Effectuez des tests de balayage et de saturation à différentes tailles au sein de chaque famille afin de déterminer le point idéal où l'utilisation et les performances sont cohérentes.

Vous devez également évaluer l'impact sur les performances de votre jeu lorsqu'il est hébergé à l'aide de conteneurs et de fonctions Lambda. Pour les cas d'utilisation où des processus de serveur de jeu de longue durée ne sont pas nécessaires, tels que les jeux asynchrones et les services de

backend de jeu, envisagez d'utiliser une architecture sans serveur avec Lambda, qui peut simplifier la gestion et les opérations pour les équipes chargées des opérations de jeu, tout en vous permettant de déployer plus rapidement votre jeu dans le monde entier auprès de nombreuses personnes. Régions AWS Pour connaître les meilleures pratiques sans serveur, reportez-vous au Serverless Applications Lens - Well-Architected Framework.

Étapes d'implémentation

- Comparez les serveurs de jeu sur des instances « c » optimisées pour le calcul pour les charges de travail intensives en termes de processeur, des instances optimisées en mémoire pour les tâches gourmandes en mémoire et des instances « n » optimisées pour le réseau pour un débit réseau élevé.
- Utilisez une mise en réseau améliorée avec Elastic Network Adapter (ENA) sur les instances prises en charge afin d'améliorer les performances du réseau, de réduire la latence et d'augmenter les taux de traitement des paquets.
- Évaluez et testez plusieurs types d'instances, processeurs (Intel, AMD, Graviton) et options d'hébergement conteneur ou Lambda, en ajustant les solutions de calcul en fonction de l'évolution des fonctionnalités du jeu.

Pour plus d'informations, consultez Choisir la bonne stratégie de calcul pour vos serveurs de jeu mondiaux.

GAMEPERF05-BP02 Déplacer les tâches de non-latency-sensitive calcul vers des flux de travail asynchrones

Lorsque vous optimisez les performances de votre jeu, il est important de garder à l'esprit que seules certaines interactions entre le client et le backend du jeu doivent être effectuées de manière synchrone. Vous devez considérer chaque fonctionnalité du point de vue de l'expérience du joueur et déterminer si certaines interactions nécessitent des communications synchrones, qui sont bloquantes et gourmandes en ressources, ou si ces fonctionnalités peuvent être mises en œuvre de manière asynchrone. Lorsque vous implémentez des appels réseau, utilisez une approche asynchrone et non bloquante. En outre, votre backend de jeu doit également être configuré pour effectuer le travail de manière efficace en déchargeant les tâches dans les files d'attente et en donnant la priorité aux réponses rapides aux clients dans la mesure du possible.

Niveau d'exposition au risque si cette bonne pratique n'est pas respectée : élevé

Directives d’implémentation

Par exemple, la mise à jour d'un classement à la fin d'une session de joueur peut être mise en œuvre de manière asynchrone afin que le client n'ait pas à attendre la fin de la mise à jour du classement. Implémentez plutôt cela de manière asynchrone sur le client du jeu et envisagez de concevoir votre service principal de manière à transférer ces types d'opérations dans des files d'attente, comme Amazon SQS. Avec cette architecture, configurez votre backend pour qu'il accepte la demande, mettez-la en file d'attente dans SQS, ce qui permet de stocker durablement les messages pour un traitement asynchrone, et répondez rapidement au client. Lorsque la mise à jour du classement est terminée, le backend peut envoyer une mise à jour au client du jeu afin que la vue du joueur sur le classement soit mise à jour.

Sinon, le joueur peut simplement se rendre sur l'écran du classement de votre jeu pour récupérer les dernières données, ce qui peut envoyer une requête Web à votre backend pour récupérer les dernières données du cache.

Étapes d’implémentation

- Déterminez si les interactions client-backend nécessitent une communication synchrone ; mettez en œuvre des approches asynchrones et non bloquantes dans la mesure du possible pour optimiser l'utilisation des ressources.
- Utilisez Amazon SQS pour vous décharger de tâches non critiques telles que les mises à jour du classement.
- Permettez au client de récupérer les données mises à jour de manière asynchrone, par exemple en récupérant les dernières données du classement à la demande ou via des mises à jour en arrière-plan.

Ressources

- Comprendre la messagerie asynchrone pour les microservices
- Lambda - Utilisation des intégrations de services et du traitement asynchrone

Gestion des données

GAMEPERF06 : Comment gérer et analyser efficacement les journaux des serveurs de jeu et stocker différents types de données de jeu pour des performances optimales ?

Les jeux peuvent contenir des données sur les joueurs, des journaux de jeu et des journaux de serveur qui doivent être dissociés les uns des autres dans la mesure du possible. La centralisation de l'ingestion des journaux et de la gestion du cycle de vie peut être bénéfique pour vos équipes de jeu en fournissant des informations sur ce qui se passe dans le jeu et sur le serveur.

Bonnes pratiques

- GAMEPERF06-BP01 Centraliser la collecte et le stockage des journaux
- GAMEPERF06-BP02 Catégoriser et stocker les données de jeu en fonction des modèles d'accès
- GAMEPERF06-BP03 Activez le formatage et le traitement par lots efficaces des journaux
- GAMEPERF06-BP04 Implémenter des politiques de rotation et de conservation des journaux
- GAMEPERF06-BP05 Utiliser des outils de surveillance et de visualisation

GAMEPERF06-BP01 Centraliser la collecte et le stockage des journaux

Mettez en œuvre une solution centralisée de collecte et de stockage des journaux pour recueillir les journaux à partir des instances de serveurs de jeux et GameLift.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Utilisez des services tels qu'Amazon CloudWatch Logs pour collecter, surveiller et stocker les données de journal de vos serveurs et GameLift instances de jeu. CloudWatch Logs fournit une solution évolutive et entièrement gérée pour la gestion des journaux, facilitant le stockage et la récupération efficaces des données des journaux sans affecter les performances du serveur de jeu. Si vous utilisez l'agent CloudWatch Logs, prenez en compte les différents types d'installation et options de configuration, telles que la taille du lot et la durée de la mémoire tampon, afin de minimiser l'impact sur le serveur de jeu. Considérez les instances du serveur de jeu comme éphémères et réduisez la dépendance à l'égard de la journalisation localisée dans la mesure du possible. Établissez une politique centralisée pour la mise en œuvre des meilleures pratiques de journalisation.

Étapes d’implémentation

- Utilisez Amazon CloudWatch Logs pour collecter, surveiller et stocker les données des journaux à partir des instances de serveurs de jeux et GameLift pour faciliter la gestion centralisée et évolutive des journaux.

GAMEPERF06-BP02 Catégoriser et stocker les données de jeu en fonction des modèles d'accès

Classez vos données de jeu en différents types en fonction de leurs modèles d'accès et de leurs besoins de stockage.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les catégories les plus courantes incluent les données des joueurs, les sauvegardes de jeu, le stockage mondial persistant et les données analytiques.

Étapes d’implémentation

Utilisez des solutions de stockage adaptées à chaque type de données afin d'optimiser les performances et la rentabilité :

- Données des joueurs : utilisez Amazon DynamoDB, une base de données NoSQL rapide et évolutive, pour stocker les profils, les préférences et les données de progression des joueurs. L'accès à faible latence et les fonctionnalités de dimensionnement automatique de DynamoDB permettent de récupérer et de mettre à jour efficacement les données des joueurs.
- Sauvegardes de jeu : utilisez Amazon S3 pour stocker les sauvegardes de jeu et les points de contrôle. S3 offre une durabilité et une évolutivité élevées pour le stockage de grandes quantités de données de sauvegarde de jeu. Envisagez d'utiliser S3 Transfer Acceleration ou Amazon CloudFront pour accélérer le chargement et le téléchargement des sauvegardes de jeu.
- Stockage mondial persistant : pour les jeux contenant des états du monde persistants ou des données de jeu partagées, pensez à utiliser Amazon DynamoDB, Amazon ou ElastiCache Amazon MemoryDB. ElastiCache et MemoryDB fournissent un stockage des valeurs clés en mémoire tandis que DynamoDB est une base de données NoSQL sauvegardée sur SSD. Ces services fournissent un accès rapide aux données stockées, réduisant ainsi le temps nécessaire

au processus du serveur de jeu pour enregistrer l'état du jeu, ce qui améliore les performances globales du processus.

- Données analytiques : utilisez Amazon Managed Streaming pour Apache Kafka for Apache Kafka ou Kinesis Data Streams pour ingérer les flux de données provenant de vos producteurs de données de jeu. Amazon Managed Service pour Apache Flink peut être utilisé pour la transformation et l'analyse en temps réel et envoyé à Amazon Data Firehose pour traitement et livraison dans des lacs de données principaux, des entrepôts et des services d'analyse. Les directives relatives au pipeline d'analyse des jeux AWS illustrent la manière dont les services fonctionnent ensemble pour fournir des analyses en temps quasi réel et par lots.

GAMEPERF06-BP03 Activez le formatage et le traitement par lots efficaces des journaux

Configurez les processus de votre serveur de jeu pour générer des journaux dans un format structuré et analysable, tel que JSON.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Mettez en œuvre des techniques de traitement par lots de journaux afin de minimiser la fréquence des transferts de données de journal depuis vos serveurs de jeu vers le stockage centralisé des journaux. Le traitement par lots des journaux réduit la surcharge réseau et améliore les performances du serveur de jeu. Utilisez des journaux détaillés ou de niveau de débogage comme exception et non par défaut, car ils peuvent entraîner une baisse des performances et des coûts qu'il convient d'éviter dans la mesure du possible.

GAMEPERF06-BP04 Implémenter des politiques de rotation et de conservation des journaux

Établissez des politiques de rotation et de conservation des journaux pour gérer la croissance des données des journaux et optimiser l'utilisation du stockage.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : bas

Directives d’implémentation

Configurez vos serveurs de jeu pour qu'ils fassent automatiquement pivoter les journaux en fonction de leur taille ou des intervalles de temps. Définissez des politiques de conservation CloudWatch des journaux dans Amazon Logs afin d'archiver ou de supprimer automatiquement les anciennes données de journal qui ne sont plus nécessaires pour une analyse active ou un dépannage.

GAMEPERF06-BP05 Utiliser des outils de surveillance et de visualisation

Utilisez des outils de surveillance et de visualisation pour mieux comprendre les performances de votre serveur de jeu et identifier les opportunités d'optimisation.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Utilisez Amazon CloudWatch pour surveiller les indicateurs clés et configurer des alarmes pour des notifications proactives. Utilisez des outils tels qu'Amazon Managed Service for Prometheus et Amazon Managed Grafana pour collecter, interroger et visualiser les métriques de vos serveurs de jeu et de votre infrastructure. Créez des tableaux de bord informatifs pour suivre les performances, identifier les goulots d'étranglement et effectuer des optimisations basées sur les données.

Réseau et diffusion de contenu

GAMEPERF07 : Comment concevez-vous votre service de matchmaking pour optimiser les performances ?

Les compétences des joueurs, la qualité des fournisseurs de services Internet (ISP) et la répartition de la population de joueurs sont des aspects importants à prendre en compte pour optimiser les performances. Les sessions de jeu peuvent être placées sur des serveurs stratégiquement situés pour égaliser les règles du jeu et organiser un match équitable.

Bonnes pratiques

- GAMEPERF07-BP01 Définissez les seuils de latence réseau pour votre jeu
- GAMEPERF07-BP02 Exécutez un service de matchmaking distinct pour chaque mode de jeu et chaque région d'hébergement de jeux
- GAMEPERF07-BP03 Surveillez régulièrement les performances du matchmaking

- GAMEPERF07-BP04 Surveillez régulièrement les performances du réseau
- GAMEPERF07-BP05 Utiliser la technologie d'accélération réseau pour améliorer les performances sur Internet

GAMEPERF07-BP01 Définissez les seuils de latence réseau pour votre jeu

Lorsque vous développez un jeu multijoueur, vérifiez que votre infrastructure de jeu n'introduit pas de latence inutile pour les joueurs. Si votre jeu est sensible à la latence du réseau, vous devez définir des seuils de latence dans votre logique de matchmaking afin de donner la priorité au placement des joueurs sur des sessions de serveur de jeu hébergées sur des serveurs de jeu à proximité ou Régions AWS qui répondent à votre objectif d'expérience de jeu idéale.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Dans de nombreux jeux sensibles à la latence, il est courant de demander aux clients du jeu d'envoyer une requête ping à chacun des emplacements de l'infrastructure du jeu afin de recueillir des données de performance telles que la latence du réseau, l'instabilité et la perte de paquets, et de communiquer ces données au backend de collecte des métriques afin qu'elles puissent être analysées. Lorsque vous associez des joueurs à des sessions de jeu, vous pouvez configurer votre jeu pour intégrer la latence réseau perçue par le client du jeu à l'infrastructure de votre serveur de jeu comme l'une des entrées utilisées dans la logique de votre service de matchmaking.

GAMEPERF07-BP02 Exécutez un service de matchmaking distinct pour chaque mode de jeu et chaque région d'hébergement de jeux

Si votre jeu propose plusieurs modes de jeu parmi lesquels les joueurs peuvent choisir, vous devez séparer les systèmes de matchmaking pour chacun d'entre eux afin de pouvoir régler indépendamment les performances de chaque mode de jeu en fonction de ses exigences uniques et de réduire la consommation de ressources. Chaque mode de jeu aura probablement des exigences uniques en matière de latence acceptable, de taille de match, ainsi que d'autres logiques de matchmaking personnalisées spécifiques au jeu. Ils attireront également probablement différents types de joueurs. Exécutez le service de matchmaking de chaque mode de jeu en tant que déploiement logiciel distinct afin de pouvoir tester les performances et utiliser les modes de jeu indépendamment.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Par exemple, vous pouvez les exécuter en tant que fonctions Lambda distinctes pour chaque mode de jeu, ou vous pouvez les exécuter en tant que déploiements de services distincts basés sur des conteneurs.

Déployez vos services de jumelage dans plusieurs régions à proximité de l'emplacement de votre serveur de jeu. Le trafic des joueurs empruntera de nombreux itinéraires, il est donc important pour le service de matchmaking de maintenir un profil de up-to-date latence sur plusieurs ISPs afin d'améliorer l'efficacité du placement des sessions de jeu à faible latence. GameLift FlexMatch fournit des conseils supplémentaires pour la sélection des régions pour les matchmakers, et inclut la possibilité d'intégrer vos matchmakers aux files d'attente de sessions de jeu multirégionales.

GAMEPERF07-BP03 Surveillez régulièrement les performances du matchmaking

L'un des moyens les plus remarquables d'optimiser les performances d'un jeu pour les joueurs est de réduire le temps qu'ils doivent attendre avant de pouvoir entrer dans une session de jeu. Les longs temps d'attente peuvent entraîner une perte d'intérêt des joueurs et une usure, il est donc important d'en tenir compte lors de la conception de votre solution de matchmaking.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Lorsque vous concevez votre configuration de matchmaking pour votre jeu, créez des règles qui déterminent les conditions appliquées pour former un match. Vous devez tenir compte de l'impact que ces règles auront sur les performances du système, en particulier sur les temps d'attente pour les joueurs. Avant d'apporter des modifications à votre implémentation de matchmaking, telles que l'ajout de nouvelles conditions de matchmaking ou de nouveaux filtres, testez-les au préalable ou envisagez de diffuser progressivement cette modification à un petit échantillon de joueurs, sous forme de canari, ou A/B testez pour recueillir des indicateurs de performance avant de présenter cette modification à l'ensemble de la population de joueurs.

Configurez votre service de matchmaking pour générer des journaux détaillés afin de comprendre les conditions ou les règles appliquées à chaque demande de matchmaking. Cela aide à la révision et à ajuster la mise en œuvre du matchmaking si nécessaire.

Par exemple, Amazon GameLift FlexMatch fournit un service de jumelage entièrement géré qui peut être utilisé en tant que service autonome avec votre propre hébergement de serveur de jeu

ou utilisé avec des serveurs de jeux hébergés sur Amazon GameLift. FlexMatch peut générer des notifications d'événements pour Amazon EventBridge, voir Configurer les notifications FlexMatch d'événements. Utilisez Amazon Simple Notification Service (Amazon SNS) pour recevoir les données de matchmaking au format JSON, ce qui vous permet de traiter et de stocker automatiquement ces informations à des fins d'analyse afin d'améliorer les performances de matchmaking.

Configurez des métriques pour suivre le temps nécessaire à votre service de matchmaking pour trouver une session de jeu adaptée aux joueurs. Passez régulièrement en revue les indicateurs de durée du matchmaking et corrélez ces temps avec le comportement des joueurs et le sentiment de la communauté. Utilisez ces données pour développer des seuils appropriés pour les délais de matchmaking qui peuvent être inclus dans la configuration de vos règles de matchmaking.

Par exemple, Amazon GameLift FlexMatch fournit une assistance pour définir les délais d'expiration des demandes de matchmaking ainsi que pour créer des règles de matchmaking qui peuvent assouplir les exigences au fil du temps. Cette fonctionnalité vous permet de créer un matchmaking qui peut s'adapter pour faciliter la création de matchs et placer les joueurs dans des sessions de jeu lorsque les matchs sont difficiles à trouver.

GAMEPERF07-BP04 Surveillez régulièrement les performances du réseau

Pour les parties compétitives, il est important d'offrir une expérience de jeu cohérente.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Un jeu qui dure 50 ms de manière fiable pour une base de joueurs plus importante est plus juste et plus amusant qu'un match où un joueur a un ping de 10 ms et un autre de 70 ms. Les changements de routage des FAI peuvent avoir un impact sur une partie de la population de joueurs, et votre système de matchmaking devra s'adapter. Amazon CloudWatch Network Monitoring vous aide à déterminer si le problème provient de votre jeu ou du fournisseur d'accès Internet du joueur.

Étapes d’implémentation

- Utilisez Amazon Cloudwatch Network Monitoring pour suivre les performances du réseau et identifier les problèmes de routage.
- Utilisez les journaux de flux VPC pour identifier les modèles de trafic anormaux ou les paquets perdus, ce qui peut indiquer une congestion du réseau, des problèmes avec les fournisseurs de services Internet ou des erreurs de configuration ayant un impact sur la latence des joueurs.

GAMEPERF07-BP05 Utiliser la technologie d'accélération réseau pour améliorer les performances sur Internet

En plus de placer physiquement l'infrastructure de jeu sensible à la latence plus près des joueurs, vous pouvez également améliorer l'expérience des joueurs en optimisant les performances réseau de votre jeu. AWS utilise le protocole BGP pour influencer le routage Internet afin d'utiliser le chemin le plus rapide vers notre réseau frontalier auprès des fournisseurs de services Internet. Si vous gérez votre propre réseau et que vous avez besoin de plus de contrôle et d'observabilité sur le comportement de routage et la publicité BGP, vous pouvez utiliser le peering privé ou Direct Connect pour acheminer le trafic d'Internet vers votre jeu sur lequel vous jouez. AWS

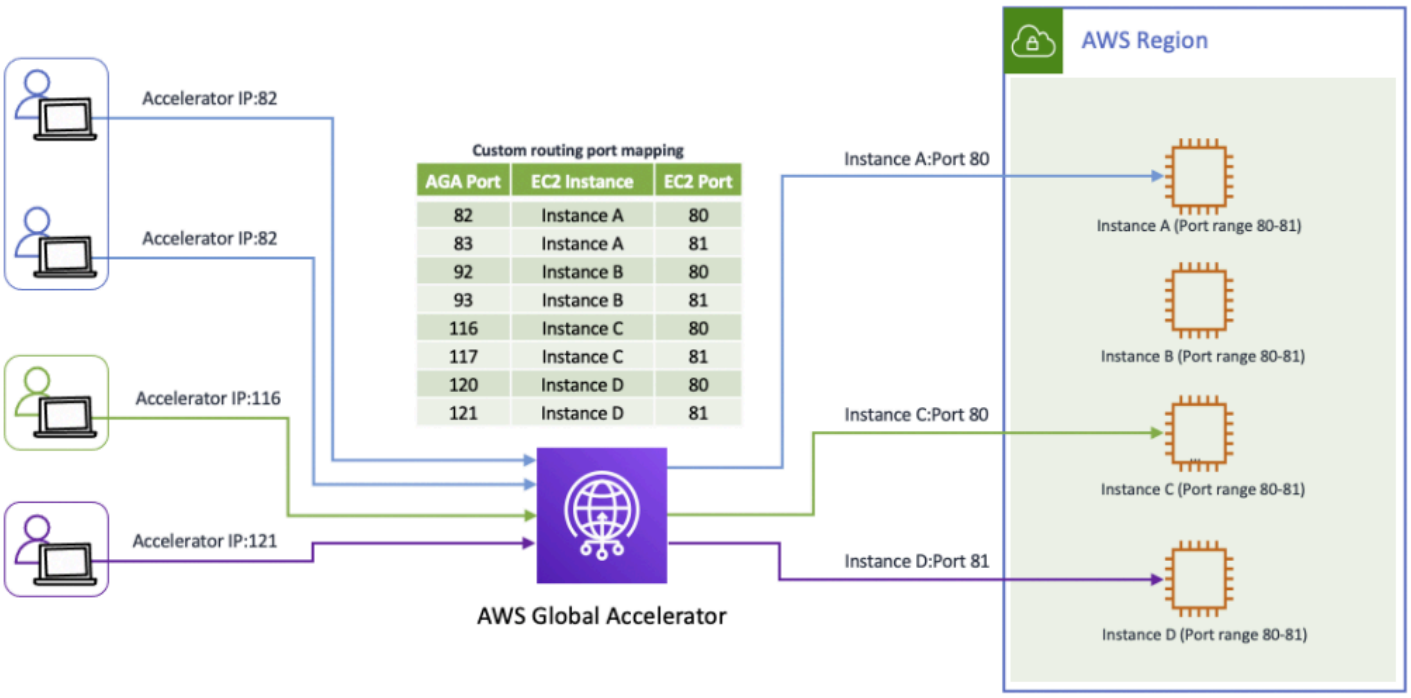
Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Tenez compte de l'architecture de référence suivante pour améliorer les performances et la réactivité d'Internet.

Performances réseau améliorées pour les jeux grâce à Global Accelerator

Pour une solution entièrement gérée de routage réseau, AWS Global Accelerator améliore les performances réseau de votre application en utilisant le réseau AWS mondial, qui peut être utilisé pour accélérer le trafic de jeu, le chat vocal et le trafic de messagerie en temps réel, ainsi que d'autres applications sensibles à la latence, tout en fournissant un basculement rapide vers vos serveurs de jeu. Les accélérateurs de routage personnalisés de Global Accelerator peuvent être intégrés à votre service de matchmaking pour fournir un routage déterministe de plusieurs joueurs vers la même session de jeu en utilisant des adresses IP et des ports anycast statiques.



Vos équipes de développement de jeux peuvent être réparties dans le monde entier et ont besoin d'un accès performant au contenu ou aux ressources partagés. Pour améliorer les performances du contenu partagé stocké dans les compartiments Amazon S3, vous pouvez configurer la réplication bidirectionnelle de vos données entre les régions à l'aide de la réplication entre régions S3 afin que les utilisateurs puissent accéder aux données depuis les compartiments les plus proches de chez eux. Pour simplifier ce modèle d'accès, utilisez les points d'accès multirégionaux S3 qui accélèrent les demandes adressées à S3 sur le réseau mondial à l'aide de Global Accelerator.

Pour plus d'informations, consultez Améliorer l'expérience des joueurs en tirant parti de AWS Global Accelerator et d'Amazon GameLift FleetIQ.

Étapes d’implémentation

- Utilisez AWS Global Accelerator pour améliorer les performances du réseau en matière de trafic de jeu, de chat vocal et de messagerie en temps réel, tout en facilitant le basculement rapide vers les serveurs de jeu.
- Configurez les accélérateurs de routage personnalisés de Global Accelerator pour les intégrer à votre service de matchmaking, permettant le routage déterministe des joueurs vers les sessions de jeu en utilisant un anycast statique. IPs
- Activez la réplication interrégionale S3 pour répliquer le contenu partagé entre les régions pour les équipes de développement de jeux distribuées.

- Utilisez les points d'accès multirégionaux S3 pour accélérer l'accès aux données S3 sur le réseau AWS mondial pour les utilisateurs répartis dans le monde entier.

Processus et culture

GAMEPERF08 : Comment adaptez-vous la barre de performance de votre jeu aux attentes des joueurs et des développeurs ?

Connaître votre joueur et votre développeur est l'un des aspects les plus importants pour améliorer l'efficacité des performances. Proposer un jeu performant avec une faible charge opérationnelle est l'un des meilleurs moyens de montrer aux joueurs et aux développeurs que vous vous souciez de leur expérience et que vous pouvez différencier votre jeu de votre studio.

Bonnes pratiques

- GAMEPERF08-BP01 Informez et incluez le joueur dans votre processus
- GAMEPERF08-BP02 Alignez le choix de la solution avec les compétences et l'expertise de l'équipe d'ingénierie

GAMEPERF08-BP01 Informez et incluez le joueur dans votre processus

Fournissez une option permettant d'afficher dans le jeu des statistiques telles que la latence, le nombre d'images par seconde et les paquets perdus. Identifiez les problèmes d'infrastructure et les interruptions de maintenance par le biais de communications avec les joueurs, telles que des pages d'état. Célébrez les nouveaux lieux de jeu grâce à des communications avec les joueurs, notamment des blogs de développement, et définissez vos attentes en matière d'amélioration de l'expérience des joueurs.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Inclure le joueur

Proposez un processus simple de soumission de diagnostic qui collecte les fichiers pertinents et les joint à un ticket d'assistance aux joueurs envoyé par votre client de jeu. Activez un forum d'assistance où les joueurs peuvent s'entraider et participer à l'amélioration de l'expérience de jeu

Tenez compte des compromis par rapport aux attentes des joueurs

Le déplacement des systèmes principaux pour des raisons de rentabilité peut ne pas être perceptible pour les joueurs, mais le déplacement des serveurs de jeu peut modifier le temps de réponse. Faites preuve de cohérence et d'équité envers les joueurs en justifiant l'extension et la réduction de vos sites d'hébergement de jeux.

Les communautés de joueurs et les zones géographiques auront leurs propres caractéristiques qui peuvent avoir un impact sur les attentes à l'égard de votre jeu. Par exemple, la Corée du Sud possède l'un des réseaux Internet les plus rapides de la planète et l'on s'attend à une latence à un chiffre, ce qui favorise un jeu très compétitif. Le gameplay occasionnel sur les appareils mobiles crée un profil de performance et un schéma d'utilisation différents par rapport au jeu de session sur console ou PC.

La connexion et le lobby font partie de l'expérience et devraient être réactifs, même si le serveur est hors ligne pour des raisons de maintenance. Planifier une nuit de raid ou passer du temps dans le lobby fait partie de l'expérience du joueur et il est important d'en tenir compte lors du choix des domaines prioritaires en termes d'efficacité des performances. Les joueurs peuvent laisser votre client de jeu ouvert pendant des mois, parfois ils peuvent simplement se connecter de temps en temps pour lire les notes de mise à jour. Un jeu Live Ops doit tenir compte de l'expérience globale du joueur dans le cadre du processus d'ingénierie et de la culture.

Étapes d’implémentation

- Fournissez des indicateurs intégrés au jeu tels que la latence, le nombre d'images par seconde et la perte de paquets, et communiquez les problèmes d'infrastructure et les calendriers de maintenance via des pages d'état et des mises à jour destinées aux joueurs.
- Implémentez une fonctionnalité de vidage et de soumission de diagnostics dans le client du jeu et créez un forum d'assistance pour favoriser le dépannage et l'amélioration au niveau de la communauté.
- Adaptez les optimisations des performances aux attentes de la communauté des joueurs, telles qu'une faible latence pour les régions compétitives ou des login/lobby expériences réactives pour les joueurs occasionnels et les joueurs de longue durée.
- Concevez des flux de travail Live Ops qui tiennent compte de l'ensemble de l'expérience du joueur, du gameplay actif au comportement inactif des clients, pour un engagement sans faille.

GAMEPERF08-BP02 Alignez le choix de la solution avec les compétences et l'expertise de l'équipe d'ingénierie

Évaluez les compétences et l'expertise de votre équipe en matière de gestion et d'optimisation des performances des serveurs de jeux lors du choix de votre option d'hébergement. Les solutions auto-hébergées telles que EC2 les conteneurs nécessitent une meilleure connaissance de la gestion de l'infrastructure, du réglage des performances et de la mise à l'échelle. Si votre équipe ne dispose pas de ces compétences, un service géré comme celui-ci GameLift peut être plus approprié, car il élimine de nombreuses complexités et permet à votre équipe de se concentrer sur des optimisations spécifiques au jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

En évaluant ces facteurs et en effectuant des tests de performance sur différentes options d'hébergement, vous pouvez sélectionner la solution la plus adaptée aux exigences spécifiques de votre jeu tout en optimisant l'efficacité des performances.

Ressources

Apprenez-en davantage sur nos meilleures pratiques en matière d'efficacité des performances.

Documents connexes :

- Centre d’architecture AWS
- Pilier de performance et d'efficacité — AWS Well-Architected Framework
- Comparaison de vos modèles de stockage sur site avec les services AWS de stockage
- Stockage d'instances Stockage par blocs temporaire pour les EC2 instances
- Collectez des métriques, des journaux et des traces à l'aide de l' CloudWatch agent
- CloudWatchAgent
- Comment activer et configurer la mise en réseau améliorée sur mes EC2 instances ?
- Améliorer l'expérience des joueurs en tirant parti de AWS Global Accelerator et d'Amazon GameLift FleetIQ
- Blog technologique de Riot Games : évolutivité et tests de charge pour Valorant
- Jeux en ligne à grande échelle avec une solution hybride AWS

- Méthode USE (saturation et erreurs d'utilisation)
- Instances Amazon EC2 Spot
- Des performances à grande échelle avec Amazon ElastiCache
- Stratégies de mise en cache de base de données utilisant Redis
- Amazon Virtual Private Cloud Connectivity Options
- Modèles de conception basés sur les meilleures pratiques : optimisation des performances d'Amazon S3

Benchmarks associés :

- Évaluez les volumes Amazon EBS
- Bande passante réseau de l'instance Amazon EC@

Outils associés :

- Unreal Engine : tester et optimiser votre contenu
- Profileur Unity
- Système de qualité Open 3D Engine (O3DE)
- Surveillance des GameLift serveurs Amazon
- Boîte à outils GameLift de test Amazon

Vidéos connexes :

- AWS re:INVENT 2019 : [REPEAT 2] Amazon EC2 Foundations (CMP211-R2)
- AWS re:Invent 2019 : Au service d' EC2Amazon de nouvelle génération : plongée en profondeur dans le système Nitro (03-R2) CMP3
- Commencer à utiliser Amazon GameLift FleetIQ — Discussions techniques en ligne AWS
- Zach Blitz de Riot Games parle de l'utilisation AWS pour améliorer les jeux
- AWS re:Invent 2023 - AWS Graviton : le meilleur rapport qualité/prix pour vos AWS charges de travail () CMP334

Formations associées :

- Hébergement du serveur de jeu sur AWS

- Utilisation d'Amazon GameLiftFleet IQ pour les serveurs de jeu
- Commencer à AWS utiliser for Games — Partie I
- Hébergement du serveur de jeu sur AWS
- AWS re:Invent 2023 — AWS Graviton : le meilleur rapport qualité/prix pour vos AWS charges de travail () CMP334

Optimisation des coûts

Le pilier de l'optimisation des coûts inclut le processus continu de raffinement et d'amélioration d'un système tout au long de son cycle de vie. Ce processus s'étend de la conception initiale de votre première preuve de concept à l'exploitation continue des charges de travail de production. En adoptant les pratiques décrites dans ce paper, vous pouvez créer et exploiter des systèmes sensibles aux coûts qui permettent d'obtenir les résultats commerciaux souhaités au prix le plus bas. La mise en œuvre de ces pratiques d'optimisation des coûts peut permettre à votre entreprise de maximiser la valeur de votre investissement dans le cloud.

Les jeux sont des projets créatifs uniques qui doivent rivaliser pour attirer l'attention des joueurs et gagner du temps de jeu. Avant le lancement, les développeurs de jeux n'ont souvent pas une idée précise de la popularité ou de la durabilité de leur jeu. En fonction de la stratégie de monétisation du jeu, des priorités commerciales et de l'étape du cycle de vie, les développeurs devront faire des compromis lors de l'évaluation des décisions d'optimisation des coûts.

Par exemple, pendant la phase de pré-lancement d'un nouveau jeu très attendu, l'accent est généralement mis sur le speed-to-market développement des fonctionnalités et les performances. La priorité est de vérifier que l'infrastructure peut évoluer pour répondre à la demande maximale des joueurs. À l'inverse, si un jeu échoue ou si le développement ralentit, l'accent peut être mis sur la réduction des coûts autant que possible afin de continuer à faire fonctionner le jeu pour les joueurs existants.

De nombreux développeurs de jeux exploitent également plusieurs jeux simultanément, ce qui nécessite des considérations supplémentaires. Les ressources telles que l'infrastructure, les logiciels et le personnel peuvent être partagées entre plusieurs jeux en direct, ce qui permet de compenser les pertes d'un jeu par les bénéfices d'un autre. Dans ce scénario, l'accent mis sur l'optimisation des coûts peut améliorer les finances de l'ensemble du portefeuille de jeux.

Compte tenu des modèles commerciaux uniques, de l'échelle et de l'imprévisibilité des jeux, les questions clés suivantes peuvent guider les décisions d'optimisation des coûts :

- Comment puis-je mesurer le coût d'infrastructure par joueur, par système et par fonctionnalité de jeu ?
- Quel est le juste équilibre entre l'optimisation des coûts et l'expérience du joueur pour l'étape actuelle du cycle de vie de mon jeu ?
- Comment puis-je maximiser le retour sur investissement en utilisant le modèle de tarification adapté à mes AWS ressources ?

L'application de ces meilleures pratiques et le fait de poser les bonnes questions peuvent aider les développeurs de jeux à créer et à exploiter des systèmes sensibles aux coûts qui permettent d'obtenir des résultats commerciaux tout en minimisant les coûts.

Domaines d'intérêt

Principes de conception

Outre les principes de conception issus du pilier d'optimisation des coûts du Well-Architected Framework, les principes de conception suivants optimisent les coûts liés à l'exécution de votre charge de travail de jeu dans le cloud.

- Mesurez le coût d'infrastructure par joueur, par système et par fonctionnalité de jeu : comprenez et suivez les coûts d'infrastructure requis pour des expériences et des fonctionnalités spécifiques aux joueurs sur tous les systèmes de jeu. Cela permet d'identifier les domaines de votre architecture susceptibles de nécessiter une optimisation des coûts.
- Évaluez le compromis entre l'optimisation des coûts et l'expérience du joueur : évaluez à quel stade se trouve votre jeu pour déterminer le bon objectif : expérience du joueur ou optimisation des coûts. Généralement, une fois qu'un jeu atteint une masse critique et que la population de joueurs se stabilise, il est temps de se concentrer sur l'optimisation des coûts d'exploitation. L'essentiel est de trouver un équilibre entre l'offre d'une expérience de jeu exceptionnelle et la gestion de votre infrastructure de jeu de la manière la plus rentable possible. La mise en œuvre de ces principes de conception maximise le retour sur investissement dans les jeux.

Pratiques de gestion financière du cloud

Il n'existe pas de bonnes pratiques de gestion financière dans le cloud spécifiques à Games Lens. Reportez-vous au pilier d'optimisation des coûts du Well-Architected Framework pour obtenir des conseils sur la gestion financière dans le cloud.

Sensibilisation aux dépenses et à l’utilisation

GAMECOST01 : Comment mesurez-vous le coût de vos environnements de jeu ?

Comprenez le coût par joueur, les fonctionnalités du jeu et l'environnement afin de pouvoir gérer et prévoir vos dépenses au fur et à mesure que le nombre de joueurs évolue au fil du temps et que des fonctionnalités sont ajoutées et améliorées. Tenez compte des meilleures pratiques suivantes pour gérer les coûts liés à vos différents environnements de jeu.

Bonnes pratiques

- GAMECOST01-BP01 Implémenter l'attribution du coût par joueur, fonctionnalité du jeu et environnement
- GAMECOST01-BP02 Découvrez les opportunités d'optimisation

GAMECOST01-BP01 Implémenter l'attribution du coût par joueur, fonctionnalité du jeu et environnement

L'attribution des coûts pour les serveurs de jeux est généralement plus simple à effectuer que les services de backend de jeu, car un serveur de jeu est généralement optimisé pour pouvoir héberger un nombre spécifique de joueurs simultanés par instance, qui peut être amorti en fonction du coût de fonctionnement de l'instance.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Pour les services de backend de jeu, il est recommandé de dissocier les composants de votre jeu en fonctionnalités distinctes qui peuvent être gérées comme des ressources logiques ou physiques distinctes afin de faciliter l'analyse des coûts.

Par exemple, bien qu'il puisse sembler simple d'implémenter une seule application monolithique pour héberger les services principaux de jeu, il est difficile de calculer le coût total par joueur et par fonctionnalité de jeu au fil du temps lorsque vous ajoutez de nouvelles fonctionnalités, car les coûts de calcul, de mise en réseau et de stockage des ressources sont partagés entre les fonctionnalités. Envisagez d'adopter une architecture sans serveur pour les services principaux de votre jeu avec des services tels qu'Amazon API Gateway AWS Lambda et/ou AWS Fargate pour le calcul, Amazon SQS et Amazon SNS pour la messagerie, Amazon S3 pour le stockage d'objets et Amazon DynamoDB pour le stockage de bases de données. Ces services ne sont que quelques exemples de produits dont la tarification est basée sur l'utilisation et principalement déterminée par le volume de demandes, afin que les coûts puissent être visualisés de manière précise. Des ressources individuelles telles que les fonctions Lambda, les services Fargate, les tables DynamoDB et les compartiments S3 peuvent être associées à des balises de répartition des coûts afin que vous puissiez attribuer les coûts de ces services avec des noms de fonctionnalités de jeu qui vous permettent de comprendre facilement les coûts de chacun de vos services.

Il est également recommandé de gérer séparément chacun de vos environnements de développement de jeux afin de pouvoir attribuer les coûts aux différents environnements. Généralement, les développeurs de jeux gèrent des environnements distincts pour les environnements de développement, de test, de mise en scène et de production, comme décrit dans le pilier des opérations de ce point de vue de l'industrie du jeu vidéo. Chaque environnement a généralement des exigences d'évolutivité, de performance et d'utilisation différentes et peut être géré par des équipes distinctes. Pour contrôler les coûts, organisez ces environnements de manière à pouvoir surveiller et attribuer correctement les coûts de chaque environnement.

Pour plus d'informations, consultez la documentation suivante :

- Création d'un jeu multijoueur sans serveur évolutif
- Serveurs de session de jeu autonomes avec un backend WebSockets basé
- Serveurs de session de jeu autonomes avec un backend sans serveur

Étapes d’implémentation

- Découplez les services de backend de jeu en fonctionnalités distinctes à l'aide d'architectures sans serveur ou conteneurisées telles qu'Amazon API AWS Lambda Gateway, et AWS Fargate pour permettre une attribution granulaire des coûts par fonctionnalité.

- Appliquez des balises de répartition des coûts à des ressources individuelles (par exemple, des fonctions Lambda, des tables DynamoDB et des compartiments S3) pour associer les coûts à des fonctionnalités de jeu spécifiques afin de mieux analyser les coûts.
- Gérez des environnements distincts pour le développement, les tests, le développement et la production, en organisant et en surveillant leurs coûts de manière indépendante afin de répondre aux exigences d'évolutivité et d'utilisation.

GAMECOST01-BP02 Découvrez les opportunités d'optimisation

Les développeurs et éditeurs de jeux peuvent utiliser AWS FinOps des pratiques pour optimiser leurs coûts liés au cloud et obtenir une meilleure visibilité de leurs dépenses dans le cloud. Ce faisant, les producteurs de jeux peuvent aligner le coût moyen requis pour maintenir l'infrastructure pour les joueurs sur les résultats financiers fournis par le jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : bas

Directives d’implémentation

AWS propose une solution prête à l'emploi pour la gestion financière dans le cloud afin de gérer et d'optimiser vos dépenses en matière de services cloud. Cette fonctionnalité inclut une visibilité granulaire et une analyse des coûts et de l'utilisation pour faciliter la prise de décision sur des sujets tels que les tableaux de bord des dépenses, l'optimisation, les limites de dépenses, le remboursement, ainsi que la détection et la réponse aux anomalies. Les conseils relatifs à la solution Cloud Financial Management incluent des fonctionnalités de budget et de prévision, vous fournissant une architecture définie et optimisée en termes de coûts pour vos charges de travail, afin que vous puissiez sélectionner le bon modèle de tarification et attribuer les coûts des ressources pertinents à vos équipes. Cela active les techniques de suivi, de notification et d'optimisation des coûts dans l'ensemble de votre environnement et de vos ressources. Vous pouvez gérer de manière centralisée les informations sur les dépenses et permettre aux parties prenantes critiques d'y accéder selon les besoins pour une visibilité ciblée et pour soutenir la prise de décision.

Un autre FinOps outil clé est le Cost Optimization Hub, qui fournit une vue centralisée des recommandations et des opportunités d'optimisation des coûts dans l'ensemble de votre Comptes AWS pays Régions AWS, afin que vous puissiez tirer le meilleur parti de vos AWS dépenses. Vous pouvez utiliser le Cost Optimization Hub pour identifier, filtrer et agréger les recommandations d'optimisation des AWS coûts sur votre Comptes AWS territoire Régions AWS. Il formule des recommandations sur le redimensionnement des ressources, la suppression des

ressources inactives, les Savings Plans et les instances réservées. Avec un tableau de bord unique, vous évitez d'avoir à consulter plusieurs AWS produits pour identifier les opportunités d'optimisation des coûts.

Si vos équipes de jeu utilisent Comptes AWS le partage, MyApplications in AWS Management Console Home peut être utilisé pour consulter les coûts des ressources des applications pour les charges de travail individuelles. Cette vue détaillée vous permet d'identifier les tendances de coûts spécifiques au sein de votre infrastructure de jeu, ce qui vous permet de prendre des décisions éclairées concernant l'allocation et l'optimisation des ressources.

En outre, l'examen régulier de vos données de facturation et de gestion des coûts avec AWS Data Exports permet de découvrir des opportunités de réduction des coûts cachés. Ce rapport détaillé fournit une ventilation complète de vos dépenses dans le cloud, vous permettant d'identifier les domaines dans lesquels les dépenses sont excessives, les ressources inutilisées et les opportunités de tirer parti de services ou de modèles de tarification plus rentables.

En adoptant FinOps les principes et en tirant parti des outils fournis AWS, les développeurs et éditeurs de jeux peuvent utiliser au mieux leurs ressources cloud, améliorant ainsi leurs résultats financiers et libérant des fonds pour le développement et l'innovation dans le domaine des jeux.

Étapes d’implémentation

- Utilisez les outils de gestion AWS Cloud financière pour une visibilité granulaire et détaillée, des tableaux de bord des dépenses, la détection des anomalies et l'attribution des coûts afin d'optimiser et de suivre efficacement les dépenses liées au cloud.
- Utilisez le Cost Optimization Hub pour centraliser les recommandations relatives au redimensionnement, aux Savings Plans et aux instances réservées par région et par région Comptes AWS .
- Passez régulièrement en revue les données AWS de facturation à l'aide des MyApplication exportations de données AWS afin d'analyser les coûts spécifiques à la charge de travail, de découvrir des opportunités d'économies et d'optimiser l'allocation des ressources.

Ressources rentables

GAMECOST02 : Comment choisissez-vous la bonne solution informatique pour vos serveurs de jeux ?

L'un des aspects les plus uniques de la charge de travail d'un jeu, par rapport aux autres types de charges de travail, est le serveur de jeu. Le serveur de jeu est essentiel à l'expérience des joueurs, car les joueurs s'y connectent depuis leur client de jeu pour jouer à une session de jeu.

Le serveur de jeu est également l'un des principaux facteurs de coût liés à l'exploitation d'un jeu multijoueur. Il est donc important d'optimiser la façon dont vous utilisez l'infrastructure informatique de vos serveurs de jeux afin de réduire les coûts.

Bonnes pratiques

- GAMECOST02-BP01 Optimisez le coût du transfert de données sur Internet
- GAMECOST02-BP02 Optimisez le nombre de sessions de jeu hébergées sur chaque instance de serveur de jeu pour optimiser les coûts
- GAMECOST02-BP03 Sélectionnez l'option de tarification informatique appropriée pour réduire les coûts

GAMECOST02-BP01 Optimisez le coût du transfert de données sur Internet

Bien que les frais soient AWS principalement liés au transfert de données sortantes (de sortie) de vos AWS ressources vers Internet, les sociétés de jeux vidéo peuvent être confrontées à des coûts élevés liés au transfert de données via AWS Direct Connect les équilibreurs de charge AWS Gateway, qui peuvent facturer à la fois les données entrantes (entrées) et sortantes. Mettez en œuvre des solutions qui réduisent le coût global du transfert des données du AWS backend de votre jeu à vos joueurs, en vous concentrant sur la minimisation des frais de sortie sur vos AWS ressources et en évaluant les options de gestion des frais d'entrée et de sortie par le biais de services de connectivité. AWS

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Utilisez Amazon CloudFront pour réduire les coûts liés à la diffusion de contenu et aux applications Web destinées au public.

Le contenu de jeu et les actifs stockés dans le cloud sont généralement stockés dans Amazon S3 et transmis au client du jeu soit directement depuis S3, soit depuis des serveurs Web hébergés sur Amazon EC2 qui récupèrent le contenu d'Amazon S3 et le transmettent aux clients. Pour réduire les coûts de transfert de données liés aux téléchargements de contenu, pensez à utiliser Amazon CloudFront devant votre espace de stockage dans le cloud pour diffuser du contenu aux utilisateurs.

L'utilisation CloudFront peut réduire le coût du transfert de données, car la diffusion de votre contenu coûte moins cher CloudFront points-of-presence que directement depuis les régions, et CloudFront elle ne facture pas de frais de récupération pour les sources AWS basées, telles qu'Amazon EC2 et Amazon S3. Si votre contenu est statique et ne change pas souvent, vous pouvez l'utiliser CloudFront pour mettre ces données en cache au plus près des utilisateurs finaux, ce qui peut réduire encore les coûts.

CloudFront améliore également la rentabilité des applications et services Web destinés au public en façade, même si la mise en cache n'est pas utilisée, car le coût du transfert de données entre vos serveurs et vos clients peut être réduit en acheminant le trafic via le réseau. AWS

Amazon CloudWatch peut être utilisé pour surveiller votre CloudFront utilisation d'Amazon. Dans les cas d'utilisation où vous utilisez plusieurs réseaux de diffusion de contenu (CDNs), Amazon CloudFront Origin Shield peut fournir une couche de mise en cache supplémentaire afin de consolider et de réduire le nombre de demandes d'origine provenant de différents fournisseurs.

Pour comprendre le trafic réseau de votre jeu, vous pouvez activer les journaux de flux VPC et Amazon CloudWatch Internet Monitor pour avoir end-to-end une visibilité sur les connexions des joueurs ou des backend du jeu. Cette approche permet d'identifier les causes du coût élevé du transfert de données et d'apporter des modifications architecturales afin d'optimiser les dépenses de transfert de données.

Étapes d’implémentation

- Utilisez Amazon avant Amazon CloudFront S3 ou les sources de contenu EC2 basées sur Amazon afin de réduire les coûts de transfert de données en tirant parti de la livraison à moindre coût CloudFront points-of-presence et en supprimant les frais de récupération de l'origine.
- Activez les journaux de flux VPC et Amazon CloudWatch Internet Monitor pour analyser le trafic réseau et identifier les modifications architecturales afin d'optimiser les coûts de transfert de données.
- Mettez en œuvre CloudFront Origin Shield pour consolider et réduire les demandes d'origine lorsque vous en utilisez plusieurs CDNs afin de réduire les coûts.

Pour en savoir plus sur les meilleures pratiques en matière de diffusion de contenu, consultez le livre blanc sur la diffusion de contenu pour les jeux.

GAMECOST02-BP02 Optimisez le nombre de sessions de jeu hébergées sur chaque instance de serveur de jeu pour optimiser les coûts

Optimisez le nombre de sessions de jeu hébergées par instance de serveur pour optimiser l'utilisation du calcul et réduire les coûts d'infrastructure informatique.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Pour optimiser les coûts, les développeurs de jeux doivent maximiser le nombre de sessions de jeu hébergées sur le même serveur physique ou virtuel, également appelé densité de compression de leurs serveurs de jeu. Cela est possible en augmentant le nombre de processus de serveur de jeu pouvant être hébergés simultanément sur une instance.

Un processus de serveur de jeu unique ne devrait généralement pas nécessiter l'utilisation de toutes les ressources disponibles sur l' EC2 instance. Il s'agit de l'un des moyens les plus importants de réduire les coûts de calcul d'un jeu et nécessite l'utilisation d'un logiciel capable de générer et de gérer plusieurs processus de serveur sur l' EC2 instance sur des ports distincts.

Par exemple, Amazon GameLift impose un quota sur le nombre maximum de processus de serveur de jeu par instance, que vous devez vous efforcer d'utiliser afin de réduire les coûts d'hébergement. Pour plus d'informations, consultez la section Points de terminaison et quotas Amazon GameLift Servers pour plus de détails sur le quota actuel pour le nombre maximal de processus de serveur de jeu par instance.

Au lieu de déployer des processus de serveur de jeu sur des machines virtuelles telles que des EC2 instances, il est de plus en plus courant pour les développeurs de jeux d'exécuter leurs serveurs de jeux sous forme d'applications basées sur des conteneurs à l'aide de solutions d'orchestration de conteneurs. Les développeurs de jeux peuvent utiliser Amazon Elastic Container Service (Amazon ECS) ou Guidance for Game Server Hosting Using Agones et Open Match sur Amazon EKS. Une autre option est Game Server Hosting on AWS Fargate, un moteur de calcul sans serveur qui fonctionne à la fois avec ECS et EKS, ce qui vous permet de vous concentrer sur votre jeu sans avoir à gérer l'infrastructure sous-jacente.

Les solutions de conteneur fournissent une fonctionnalité de planification des tâches qui permet de trouver automatiquement une instance de conteneur disponible dans le cluster pour héberger le conteneur de votre serveur de jeu en fonction des besoins en ressources et d'autres logiques de placement que vous spécifiez. Cependant, il est important de réfléchir à la manière dont vous

allez gérer le comportement de mise à l'échelle et de placement des joueurs de manière à ne pas perturber les sessions actives des joueurs.

Étapes d’implémentation

- Augmentez la densité d'emballage en exécutant plusieurs processus de serveur de jeu par EC2 instance à l'aide de ports et de logiciels de gestion des processus distincts.
- Utilisez Amazon GameLift ou des solutions de conteneurs comme ECS, EKS, ou AWS Fargate pour gérer efficacement les processus des serveurs de jeu et réduire les coûts d'infrastructure.
- Surveillez en permanence l'utilisation des ressources pour affiner la densité d'emballage et maintenir la rentabilité sans compromettre l'expérience des joueurs.

GAMECOST02-BP03 Sélectionnez l'option de tarification informatique appropriée pour réduire les coûts

Testez les performances du logiciel de votre serveur de jeu sur différents types d'instances et options de calcul afin de déterminer l'option la plus rentable pour votre jeu.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Outre l'utilisation efficace des types d' EC2 instances adaptés à votre charge de travail, déterminez laquelle des options de tarification de calcul disponibles est la plus adaptée à vos objectifs d'optimisation des coûts. Plusieurs options de tarification sont disponibles, notamment les instances à la demande, les instances ponctuelles, les instances réservées et les Savings Plans.

Les Savings Plans (SPs) offrent des remises sur le calcul en prenant des engagements d'utilisation et sont idéaux pour les scénarios dans lesquels vous ne pouvez pas prévoir votre consommation prévue sur une période d'un an ou de trois ans. Ils proposent des remises telles que les instances réservées, avec la possibilité d'appliquer ces remises en fonction des régions, de la famille d'instances, du système d'exploitation et de la location. Ils peuvent également être appliqués à AWS Fargate qui peut être une option d'hébergement de serveurs de jeux pour les jeux occasionnels ou AWS Lambda qui est utilisée comme une excellente option pour les jeux au tour par tour qui ne nécessitent pas de serveurs de jeu. Pour plus d'informations, consultez Création d'un jeu multijoueur sans serveur évolutif.

Les Savings Plans sont introduits lors des lancements de jeux afin de réduire les coûts liés à la charge de travail des serveurs de jeux, qui contribue aux dépenses d' EC2 instances lors de la sortie

du jeu auprès du public. Les Savings Plans peuvent également être introduits après le lancement lorsque l'équipe chargée des opérations du jeu aura une meilleure idée du trafic de joueurs après une période de production prolongée du jeu.

Étant donné que les Savings Plans offrent une flexibilité régionale, ils sont particulièrement idéaux pour optimiser les dépenses des serveurs de jeux pour des jeux dont l'utilisation est imprévisible d'une zone géographique à l'autre.

Par exemple, si votre mode d'utilisation quotidien des joueurs nécessite au moins 20 serveurs pour soutenir votre base de joueurs, mais qu'il en nécessite régulièrement jusqu'à 40, envisagez de souscrire des engagements Savings Plan pour couvrir les 20 serveurs de base, car cette demande d'utilisation est prévisible et constante, et permettra d'utiliser au maximum l'engagement d'utilisation que vous avez acheté.

Optimisez l'utilisation des Savings Plans et complétez-les avec d'autres options d'achat offrant plus de flexibilité en cas de pics d'utilisation imprévisibles des serveurs de jeu, telles que les instances à la demande et les instances Spot pour réaliser des économies optimales.

Les instances Spot sont idéales pour faire fonctionner des serveurs de jeux, car elles offrent les remises informatiques les plus importantes, ne nécessitent aucun engagement d'utilisation et offrent de la flexibilité pour les types de charge de travail imprévisibles et complexes. Cependant, les instances Spot peuvent être interrompues. Elles sont donc parfaitement adaptées aux charges de travail des serveurs de jeu dont la durée de session de jeu est courte ou aux situations où la tolérance d'interruption est plus élevée.

Pour plus d'informations sur les conseils relatifs à l'exécution de serveurs de jeux utilisant Kubernetes sur Amazon EKS avec des instances EC2 Spot, consultez Comment exécuter des jeux massivement multijoueurs avec Spot à EC2 l'aide d'Aurora Serverless.

Utilisez les instances Amazon EC2 Spot pour déterminer les pools présentant le moins de risques d'interruption et permettant de réaliser des économies maximales par rapport aux tarifs à la demande.

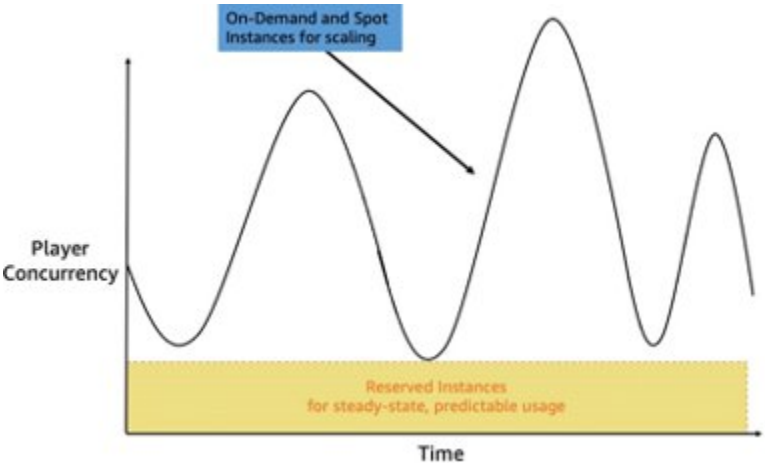
Lorsque vous utilisez Spot, il est également recommandé d'exécuter les charges de travail des serveurs de jeu sur plusieurs types d' EC2 instances et zones de disponibilité Région AWS afin de diversifier votre utilisation de la capacité et de réduire les risques d'interruption.

Envisagez d'utiliser des instances ponctuelles en combinaison avec des instances à la demande afin de minimiser l'impact des interruptions potentielles sur les sessions de jeu actives et d'utiliser une stratégie d'allocation optimisée des capacités pour réduire davantage le risque d'interruption.

Reportez-vous aux bonnes pratiques pour Amazon EC2 Spot pour en savoir plus sur les meilleures pratiques. Le rééquilibrage des capacités dans Auto Scaling pour remplacer les instances Spot à risque peut être utilisé pour surveiller de manière proactive et ajouter de la capacité supplémentaire lorsque les instances Spot présentent un risque accru d'interruption.

Amazon GameLift FleetIQ s'intègre aux instances Spot pour optimiser l'utilisation d'instances Spot à faible coût tout en réduisant le risque d'interruptions. Si vous hébergez votre jeu en utilisant GameLift, consultez la GameLift documentation relative au choix des ressources informatiques. Pour plus d'informations, voir Choisir des ressources de calcul pour un parc géré.

Le schéma suivant fournit un exemple illustrant l'utilisation de plusieurs options de tarification informatique pour les charges de travail des serveurs de jeu :



Hébergement de serveurs de jeux avec plusieurs options de EC2 tarification

Dans le diagramme, la simultanéité des joueurs fluctue au fil du temps, ce qui complique la gestion de l'utilisation et l'optimisation des coûts. Pour faire face à cette fluctuation, envisagez d'adopter une combinaison de différentes options de tarification informatique, en utilisant Savings Plans pour répondre EC2 à vos exigences d'utilisation minimale tout en vous appuyant sur les instances EC2 On-Demand et EC2 Spot pour répondre aux besoins de vos joueurs.

Étapes d’implémentation

- Utilisez Savings Plans pour une utilisation de base prévisible, en les associant à des instances ponctuelles et à la demande pour plus de flexibilité et d'optimisation des coûts lors des pics d'utilisation.
- Utilisez des instances ponctuelles pour les serveurs de jeux dont la durée des sessions est courte ou dont la tolérance aux interruptions est plus élevée, en diversifiant les types d'instances et les zones de disponibilité afin de minimiser les risques.

- Mettez en œuvre des outils tels que EC2 Spot Instances Advisor, Capacity Rebalancing et GameLift FleetIQ pour optimiser l'utilisation des instances Spot et gérer les interruptions de manière proactive.

Coûts de transfert des données

GAMECOST03 : Comment optimisez-vous les coûts de transfert de données pour votre infrastructure de jeu ?

Les jeux peuvent transférer une quantité importante de données sur Internet entre les appareils clients de jeu de vos joueurs et votre infrastructure de jeu pour offrir une expérience de jeu optimale, ainsi qu'entre les composants de votre infrastructure de jeu.

Par exemple, le transfert de données se produit lorsque les joueurs téléchargent les mises à jour du contenu du jeu sur leurs clients de jeu, enregistrent l'état de leur progression dans le cloud, participent à des sessions de jeu multijoueur en temps réel avec leurs amis et lorsque votre infrastructure de jeu transfère des données entre les régions et les zones de disponibilité. Il est important de comprendre où le transfert de données s'effectue dans votre charge de travail de jeu afin d'optimiser vos choix d'architecture afin de réduire ce coût de transfert de données.

Pour optimiser les coûts de transfert de données en fonction de votre charge de travail de jeu, tenez compte des meilleures pratiques suivantes :

Bonnes pratiques

- GAMECOST03-BP01 Choisissez le type de stockage approprié pour le contenu généré par les utilisateurs afin de réduire les coûts
- GAMECOST03-BP02 Optimiser les bases de données pour les backends de jeu

GAMECOST03-BP01 Choisissez le type de stockage approprié pour le contenu généré par les utilisateurs afin de réduire les coûts

Chaque type de données généré et stocké dans votre jeu possède des caractéristiques uniques que vous devez prendre en compte pour déterminer la solution de stockage adaptée à votre charge de travail.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : bas

Directives d’implémentation

Utilisez Amazon S3 Object Lifecycle Management pour stocker les données des objets dans la classe de stockage la plus rentable. Amazon S3 propose plusieurs classes de stockage et une gestion du cycle de vie des objets pour faciliter la mise en place de politiques simples et précises afin de transférer automatiquement les données entre les niveaux de stockage afin de réduire les coûts. Au lieu de simplement stocker les données dans la classe de stockage standard S3 par défaut, envisagez de configurer une configuration de cycle de vie pour transférer automatiquement les données entre les niveaux au fil du temps, ou utilisez la classe de stockage S3 Intelligent-Tiering pour les modèles d'accès inconnus ou changeants.

Par ailleurs, S3 Intelligent-Tiering permet de transférer automatiquement et de manière rentable les données entre les niveaux. Elle est recommandée comme classe de stockage par défaut car elle permet d'optimiser les coûts sans qu'il soit nécessaire de configurer manuellement des politiques de cycle de vie. Elle constitue désormais le meilleur choix pour les objets de petite taille et de courte durée. Pour plus d'informations, consultez Amazon S3 Intelligent-Tiering — Améliorations des coûts optimisées pour les objets de courte durée et de petite taille.

Les cas d'utilisation courants d'Amazon S3 incluent le stockage d'actifs de jeu, de contenu statique, de journaux de jeu, de stockage de lacs de données et de sauvegardes. Pour les cas d'utilisation nécessitant des systèmes de fichiers, tels que l'attachement de systèmes de fichiers partagés à des postes de travail pendant le développement, pensez à utiliser Amazon Elastic File System (Amazon EFS), qui fournit différentes classes de stockage et augmente et diminue automatiquement au fur et à mesure que vous ajoutez et supprimez des fichiers sans qu'il soit nécessaire de gérer l'infrastructure.

Amazon S3 One Zone -IA est une option de stockage idéale pour les données éphémères liées aux sessions de jeu, au matchmaking ou à d'autres informations éphémères qui peuvent être recréées selon les besoins. Ce type de données de jeu ne nécessite pas de redondance entre plusieurs zones de disponibilité ()AZs. Cette classe de stockage à moindre coût convient parfaitement aux enregistrements des actions des joueurs, aux événements du jeu et aux autres données de télémétrie utilisées à des fins d'analyse ou de débogage.

Le principal avantage en termes d'optimisation des coûts de l'utilisation de S3 Express One Zone pour de telles données de jeu réside dans les économies importantes réalisées par rapport à la classe de stockage S3 standard, avec une réduction des coûts de stockage allant jusqu'à 20 %. Cela peut être particulièrement avantageux pour les jeux contenant de gros volumes de données qui ne nécessitent pas le même niveau de durabilité et de disponibilité que les données d'applications

critiques. En tirant parti de S3 One Zone, les développeurs et éditeurs de jeux peuvent optimiser leurs coûts de stockage dans le cloud sans compromettre l'expérience globale des joueurs.

Étapes d’implémentation

- Configurez les politiques de cycle de vie d'Amazon S3 pour transférer les données entre les classes de stockage ou utilisez S3 Intelligent-Tiering par défaut pour une optimisation automatique des coûts en fonction de l'évolution des modèles d'accès.
- Utilisez S3 One Zone Infrequent Access pour les données de session de jeu transitoires, telles que les enregistrements de télémétrie et de matchmaking, afin de réduire les coûts de stockage jusqu'à 20 % tout en maintenant une disponibilité suffisante.
- Pour répondre aux besoins en matière de systèmes de fichiers partagés pendant le développement, utilisez Amazon EFS pour simplifier la gestion du stockage grâce à une capacité élastique et à plusieurs classes de stockage.

GAMECOST03-BP02 Optimiser les bases de données pour les backends de jeu

Les jeux s'appuient largement sur des bases de données pour stocker un large éventail de données critiques, qu'il s'agisse des profils et des inventaires des joueurs, des microtransactions en jeu ou des indicateurs de progression. Les bases de données jouent également un rôle crucial dans la gestion des aspects sociaux des jeux, tels que la création et le maintien de groupes de joueurs, de groupes de joueurs et l'application des politiques de modération. À mesure que le nombre de joueurs d'un jeu augmente, les coûts associés à la base de données augmentent inévitablement pour répondre aux demandes croissantes en matière de données et d'utilisation.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : moyen

Directives d’implémentation

Pour les backends de jeu exécutés sur Amazon Aurora, plusieurs stratégies d'optimisation des coûts peuvent être utilisées. L'une des principales recommandations est de dimensionner automatiquement vos répliques de lecture en fonction des modèles d'utilisation, en augmentant ou en diminuant dynamiquement le nombre de répliques pour gérer les fluctuations du trafic. Cela signifie que vous payez pour les ressources dont vous avez réellement besoin. Une autre tactique d'optimisation consiste à remplacer les répliques de lecture utilisées pour l'analyse des jeux par des exportations d'instantanés de base de données vers Amazon S3, car le service de stockage S3 est

généralement plus abordable que les instances de base de données Aurora provisionnées. Pour plus d'informations, consultez Exportation de données instantanées de base de données vers Amazon S3 pour Amazon RDS.

L'exploration de l'utilisation d'instances de base de données réservées pour Amazon Aurora pour vos instances de base de données principales et la transition vers la configuration sans serveur Aurora peuvent également permettre de réaliser des économies substantielles à long terme en offrant plus de flexibilité et un contrôle granulaire de l'utilisation de vos ressources.

De même, pour les backends de jeu qui utilisent Amazon DynamoDB, l'utilisation du mode de capacité à la demande de DynamoDB peut être un choix efficace, en particulier pour les charges de travail nouvelles ou imprévisibles, car il vous permet de ne payer que pour les ressources que vous consommez sans avoir à surapprovisionner. Au fur et à mesure que les modèles de trafic de votre jeu deviennent plus stables et prévisibles au fil du temps, vous pouvez passer au mode de capacité provisionnée DynamoDB, qui peut permettre de réaliser des économies grâce à une meilleure planification des capacités. L'activation de l'auto-scaling sur vos tables DynamoDB constitue une autre optimisation clé, permettant au service d'ajuster dynamiquement la capacité allouée en fonction des fluctuations du trafic. Testez la structure de données de votre jeu dans un environnement de développement avant de le lancer afin de trouver et de supprimer les index secondaires locaux (LSIs) et les index secondaires globaux (GSIs) inutiles. Cela peut permettre de réaliser des économies substantielles en termes de stockage et d'exploitation des données de jeu. La suppression des opérations de scan inefficaces du code de votre backend de jeu au profit de requêtes plus ciblées, l'achat de capacité réservée à Amazon DynamoDB et l'utilisation de DynamoDB Streams AWS Lambda avec des déclencheurs pour traiter les événements du backend de jeu peuvent encore optimiser vos coûts DynamoDB. Pour plus d'informations, consultez la section Meilleures pratiques en matière d'interrogation et d'analyse de données dans DynamoDB.

En mettant en œuvre ces stratégies d'optimisation des coûts pour Amazon Aurora et DynamoDB, les développeurs et éditeurs de jeux peuvent réduire de manière significative leurs dépenses liées aux bases de données de base de données de base de données de jeu.

Étapes d’implémentation

- Utilisez Aurora Read Replica auto-scaling et les exportations de snapshots de base de données vers Amazon S3 pour gérer de manière rentable les fluctuations du trafic et les besoins d'analyse.
- Optimisez les coûts DynamoDB en commençant par une capacité à la demande pour les nouvelles charges de travail, en passant à une capacité provisionnée avec auto-scaling pour un trafic prévisible, et en supprimant les et inutilisés. LSIs GSIs

- Évitez les opérations de scan inefficaces au profit de requêtes ciblées, utilisez des instances réservées ou des capacités réservées, et utilisez DynamoDB Streams pour le traitement des AWS Lambda événements.

Gestion des ressources de demande et d'offre

Il n'existe pas de bonnes pratiques de gestion de la demande et des ressources d'offre spécifiques au Games Lens.

Pour plus d'informations sur la gestion de la demande et la fourniture de ressources, voir Cost Optimization Pillar - AWS Well-Architected Framework.

Optimisation au fil du temps

Il n'existe pas de bonnes pratiques d'optimisation au fil du temps spécifiques au Games Lens.

Pour plus d'informations sur l'optimisation des coûts au fil du temps, voir Cost Optimization Pillar - AWS Well-Architected Framework.

Ressources

Consultez les ressources suivantes pour en savoir plus sur les meilleures pratiques en matière d'optimisation des coûts :

Documents connexes :

- Comment réduire les frais de transfert de données pour ma passerelle NAT dans Amazon VPC ?
- Présentation de l'adaptateur Amazon GameLift FleetIQ pour Agones
- Comment puis-je trouver les principaux contributeurs au trafic de passerelle NAT dans mon Amazon VPC ?
- Choisissez la bonne stratégie de calcul pour vos serveurs de jeu mondiaux
- AWS Well-Architected Labs — Ressources rentables
- Le plugin Amazon VPC CNI augmente le nombre de pods par nœud
- Meilleures pratiques en matière d'architecture pour l'optimisation des coûts
- Réduction du temps d'attente des joueurs et dimensionnement correct de l'allocation de calcul à l'aide d'Amazon SageMaker AI RL et d'Amazon EKS

- Optimiseur de calcul AWS
- Electronic Arts optimise les coûts et les opérations de stockage à l'aide d'Amazon S3 Intelligent-Tiering et d'Amazon Glacier
- Échappez aux pratiques de licence peu favorables en migrant les charges de travail Windows vers Linux
- Présentation des coûts de transfert des données pour les architectures courantes
- AWS et Kubecost collaborent pour fournir un suivi des coûts aux clients d'EKS
- Jeter les bases : configurer votre environnement pour optimiser les coûts
- Présentation des instances Amazon EC2 Spot

Durabilité

Le pilier du développement durable fournit des principes de conception, des conseils opérationnels, des meilleures pratiques et des plans d'amélioration pour vous aider à atteindre les objectifs de durabilité pour vos charges AWS de travail.

Vous trouverez des conseils de mise en œuvre dans le livre blanc Sustainability Pillar - AWS Well-Architected Framework.

Domaines d'intérêt

- Principes de conception
- Sélection d’une région
- Alignement sur la demande
- Logiciels et architecture
- Gestion des données
- Matériel et services
- Ressources

Principes de conception

La durabilité de l'industrie du jeu vidéo évolue à des rythmes divers dans les différentes régions du monde. Avec l'énergie durable dans de vastes régions d'Amérique du Nord et les mandats de durabilité dans l'UE et au Royaume-Uni, les architectes ont plusieurs approches pour réduire leurs charges de travail dans un futur proche. Le pilier de durabilité Well-Architected peut être utilisé pour mettre en œuvre ces mesures pour une grande variété de charges de travail.

Cette section de l'objectif décrit plusieurs bonnes pratiques qui peuvent être utilisées pour les charges de travail liées aux jeux.

- Sélectionnez le type de stockage approprié pour les données utilisateur des jeux.
- Soyez conscient des politiques de cycle de vie des données qui dédupliqueront les données et supprimeront les données inutiles de vos charges de travail.
- Soyez sélectif quant à la bonne taille du déploiement des ressources de calcul.
- Utilisez le mode sans serveur pour les processus courts et transactionnels.

Sélection d’une région

Il n'existe pas de bonnes pratiques de sélection des régions spécifiques au Games Lens. Pour plus de détails, voir Sustainability Pillar - AWS Well-Architected Framework.

Alignement sur la demande

Il n'y a aucun alignement pour exiger les meilleures pratiques spécifiques au Games Lens. Pour plus de détails, voir Sustainability Pillar - AWS Well-Architected Framework.

Logiciels et architecture

Il n'existe pas de bonnes pratiques en matière de logiciel ou d'architecture spécifiques au Games Lens. Pour plus de détails, voir Sustainability Pillar - AWS Well-Architected Framework.

Gestion des données

GAMESUS01 : Comment gérez-vous les données des utilisateurs et des jeux dans votre système de jeu ?

Développez une stratégie de cycle de vie des données qui optimise le stockage et la pertinence des données en ne conservant que les données historiques critiques.

Bonnes pratiques

- GAMESUS01-BP01 Utiliser des technologies de stockage adaptées aux modèles adaptés au contenu utilisateur, aux informations sur les abonnés et aux achats en jeu
- GAMESUS01-BP02 Utilisez des politiques de cycle de vie ou l'expiration du TTL pour supprimer les jeux, les données utilisateur, les fichiers journaux ou les actifs obsolètes inutiles

GAMESUS01-BP01 Utiliser des technologies de stockage adaptées aux modèles adaptés au contenu utilisateur, aux informations sur les abonnés et aux achats en jeu

Vous devez classer vos données par type, par besoin de conservation et par fréquence d'accès. Cela vous permet de sélectionner la solution de stockage la plus optimisée pour la myriade de types de données produits par votre jeu ou vos services principaux. Les données qui changent rapidement doivent être stockées dans des services de base de données à valeur clé ou en mémoire. Les données transactionnelles doivent être stockées dans des services de base de données relationnelle. Les fichiers volumineux, les éléments de jeu ou le contenu généré par les utilisateurs doivent être stockés dans des services de stockage d'objets.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Les jeux produisent et consomment une grande variété de types de données qui nécessitent des solutions de stockage optimisées en termes de fréquence d'accès, de latence et de coût. Les données stockées doivent être classées à l'aide de balises afin de différencier les données qui peuvent être supprimées ou qui doivent être stockées à long terme.

Les services suivants fonctionnent bien pour une variété de cas d'utilisation des Jeux :

Amazon Aurora (compatible avec MySQL et PostgreSQL) offre une haute disponibilité, une faible latence et un dimensionnement automatique, ce qui en fait un excellent choix pour gérer de grandes quantités de données transactionnelles, telles que la gestion et l'authentification des comptes des joueurs, les économies du jeu, les classements et les classements des joueurs, la persistance de l'état du jeu, la gestion des événements et des campagnes, ainsi que le déploiement multirégional et à haute disponibilité.

Amazon DynamoDB est une base de données NoSQL entièrement gérée connue pour sa faible latence, son haut débit et son évolutivité sans faille, ce qui la rend idéale pour gérer les données des joueurs en temps réel, la gestion des sessions, l'inventaire, l'économie du jeu, l'état du jeu multijoueur en temps réel, le matchmaking, la journalisation des événements et la mise à l'échelle pour un public mondial.

Amazon DocumentDB (compatible avec MongoDB) fournit un service de base de données orienté document évolutif et à faible latence, idéal pour stocker des données flexibles et semi-structurées,

telles que le système d'inventaire, les profils des joueurs et les personnalisations, les mondes de jeu et le contenu généré de manière procédurale, les interactions entre les joueurs et les réseaux sociaux, les analyses et le suivi des comportements, ainsi que les métadonnées et configurations intégrées au jeu.

Amazon ElastiCache prend en charge la mise en cache en mémoire avec Redis ou Memcached, offrant ainsi un accès rapide aux données et des temps de réponse réduits, ce qui est essentiel pour les jeux multijoueurs en temps réel où la vitesse et les performances sont essentielles pour une expérience utilisateur fluide. ElastiCache est utilisé dans les jeux vidéo pour les classements en temps réel, la gestion des sessions, la mise en cache des métadonnées du jeu, le chat et la messagerie en jeu, le matchmaking, les analyses et la télémétrie en temps réel, et la mise à l'échelle pour les événements à fort trafic.

Amazon Simple Storage Service (S3) peut être utilisé pour stocker des objets tels que des actifs de jeu, des vidéos, des photos, des fichiers journaux de texte, etc. S3 est un service de stockage d'objets offrant une évolutivité, une disponibilité des données, une sécurité et des performances de pointe.

S'il propose plusieurs classes de stockage prenant en charge l'accès fréquent et occasionnel aux données, ainsi qu'un stockage d'archives rentable. Pour les données fréquemment consultées tout au long du développement, les studios doivent stocker les objets dans le standard S3 pour une faible latence et des performances de débit élevées. Pour les données qui passent fréquemment du chaud au froid ou vice versa, les studios devraient étudier S3 Intelligent-Tiering. La hiérarchisation intelligente surveille les modèles d'accès à vos données et déplace automatiquement les données vers le niveau d'accès le plus rentable.

Pour les studios qui ont besoin d'un débit élevé, d'une faible latence et qui acceptent de vivre dans une seule zone de disponibilité, utilisez S3 Express One Zone. Cela permet de répliquer les données vers une seule AZ et d'améliorer les vitesses d'accès aux données par rapport à la norme S3. Pour les besoins d'archivage approfondis de données historiques, Amazon propose également Amazon Glacier. Les classes de stockage Amazon Glacier sont spécialement conçues pour l'archivage des données. Elles vous offrent des performances élevées, une flexibilité d'extraction et un stockage d'archives à faible coût dans le cloud.

Amazon Elastic Block Store peut être utilisé pour stocker les fichiers binaires, les fichiers exécutables et les configurations des serveurs de jeu ou des référentiels de ressources dont vos serveurs de jeux ou référentiels de ressources ont besoin pour fonctionner. Vous devez créer un instantané et supprimer les volumes inutilisés qui ne sont pas attachés à une EC2 instance. Cela vous permet de réduire les frais de stockage tout en réduisant l'utilisation de services et de matériel inutiles.

Étapes d’implémentation

- Classez les données de jeu par type, besoins de rétention et fréquence d'accès, en balisant les données pour faire la distinction entre les besoins de stockage à court terme et à long terme.
- Utilisez Amazon Aurora pour les données transactionnelles, DynamoDB pour les données des joueurs en temps réel, DocumentDB pour les données semi-structurées ElastiCache et pour la mise en cache à faible latence des informations de jeu critiques.
- Stockez les actifs de jeu, les journaux et le contenu généré par les utilisateurs dans Amazon S3, en sélectionnant les classes de stockage appropriées (par exemple, Intelligent-Tiering, One Zone et Glacier) en fonction des modèles d'accès et des besoins d'archivage, et utilisez EBS pour les binaires et les configurations des serveurs de jeux avec une gestion régulière des instantanés.

GAMESUS01-BP02 Utilisez des politiques de cycle de vie ou l'expiration du TTL pour supprimer les jeux, les données utilisateur, les fichiers journaux ou les actifs obsolètes inutiles

Vous pouvez utiliser des balises et des types de données pour créer des politiques de cycle de vie ou des TTL pour déplacer les données vers le stockage d'archives ou les supprimer complètement du service. Cela peut inclure des configurations temporaires, du contenu archivé expiré et des journaux historiques qui ne sont plus nécessaires. La plupart des services prennent en charge le balisage.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Pour les données stockées dans S3, vous pouvez utiliser des politiques de cycle de vie pour déplacer les données vers des niveaux de stockage à accès peu fréquent et d'archivage. Dans une configuration de cycle de vie S3, vous pouvez définir des règles afin d’effectuer la transition d’objets d’une classe de stockage vers une autre et économiser les coûts de stockage. Si vous ne connaissez pas les modèles d’accès de vos objets ou que vos modèles d’accès évoluent au fil du temps, vous pouvez effectuer la transition des objets vers la classe de stockage S3 Intelligent-Tiering pour réduire automatiquement les coûts.

Amazon S3 prend en charge un modèle en cascade pour la transition entre classes de stockage, comme illustré dans le schéma suivant.

Vous pouvez ajouter des actions de transition à une configuration du cycle de vie S3 afin d’indiquer à Amazon S3 de supprimer les objets à la fin de leur durée de vie. Lorsqu'un objet atteint la fin de

sa durée de vie en fonction de la configuration de son cycle de vie, Amazon S3 exécute une action d'expiration en fonction de l'état de version S3 dans lequel se trouve le compartiment :

- Compartiment non versionné : Amazon S3 met l'objet en file d'attente pour le supprimer et le supprime de manière asynchrone, le supprimant définitivement.
- Compartiment activé pour la gestion des versions : si la version actuelle de l'objet n'est pas un marqueur de suppression, Amazon S3 ajoute un marqueur de suppression avec un ID de version unique. Cela définit la version actuelle comme ancienne et le marqueur de suppression devient la version actuelle.
- Compartiment suspendu à la version : Amazon S3 crée un marqueur de suppression dont l'ID de version est nul. Ce marqueur de suppression remplace une version d'objet par un ID de version nul dans la hiérarchie des versions, ce qui supprime effectivement l'objet.
- Quand vous ajoutez une configuration de cycle de vie dans un compartiment, les règles de configuration s’appliquent à la fois aux objets existants et à ceux que vous ajouterez ultérieurement. Par exemple, si vous ajoutez une règle de configuration du cycle de vie aujourd'hui avec une action d'expiration qui fait expirer les objets dotés d'un préfixe spécifique 30 jours après leur création, Amazon S3 mettra en file d'attente pour suppression les objets existants âgés de plus de 30 jours et portant le préfixe spécifié.

La durée de vie (TTL) de DynamoDB est une méthode économique pour supprimer les éléments qui ne sont plus pertinents. Elle vous permet de définir un horodatage d’expiration par élément pour indiquer quand cet élément n’est plus nécessaire. DynamoDB supprime automatiquement les éléments ayant expiré quelques jours après leur date d’expiration, sans consommer de débit d’écriture.

- Pour utiliser la TTL, activez-la d’abord sur une table, puis définissez un attribut spécifique pour stocker l’horodatage d’expiration de la TTL. L’horodatage doit être stocké au format d’heure epoch Unix avec une granularité de quelques secondes. Chaque fois qu’un élément est créé ou mis à jour, vous pouvez calculer le délai d’expiration et l’enregistrer dans l’attribut TTL.
- Les éléments dont les attributs TTL sont valides et expirés peuvent être supprimés par le système, généralement quelques jours après leur expiration. Vous pouvez à tout moment mettre à jour les éléments ayant expiré qui sont en attente de suppression, notamment en modifiant ou en supprimant leurs attributs TTL. Lors de la mise à jour d’un élément ayant expiré, nous vous recommandons d’utiliser une expression conditionnelle pour vous assurer que l’élément n’a pas été supprimé ultérieurement. Utilisez des expressions de filtre pour supprimer les éléments ayant expiré des résultats Scan et Query.

- Les éléments supprimés fonctionnent de la même manière que ceux supprimés par le biais d'opérations de suppression classiques. Une fois supprimés, les éléments sont placés dans DynamoDB Streams sous forme de suppression de service au lieu d'être supprimés par l'utilisateur et sont supprimés des index secondaires locaux et des index secondaires globaux, comme les autres opérations de suppression.

Avec ElastiCache for Redis, vous pouvez contrôler l'actualité de vos données mises en cache en utilisant TTLs ou en expirant les clés mises en cache. Une fois le délai défini écoulé, la clé est supprimée du cache et l'accès au magasin de données d'origine est requis pour accéder aux données mises à jour.

- Deux principes déterminent les modèles appropriés TTLs à appliquer et les types de modèles de mise en cache à implémenter. Tout d'abord, il est important de comprendre le taux de variation des données sous-jacentes. Ensuite, il est important que vous évaluiez le risque que des données obsolètes soient renvoyées à votre application plutôt qu'à leur équivalent mis à jour.
- Dans le cas de données dynamiques qui changent souvent, vous souhaiterez peut-être appliquer une TTLs durée inférieure pour faire expirer les données à un rythme correspondant à celui de la base de données principale. Cela réduit le risque de renvoyer des données obsolètes tout en fournissant une mémoire tampon pour décharger les demandes de base de données.
- Il est également important de savoir que, même si vous ne mettez en cache des données que pendant des minutes ou des secondes au lieu de longues durées, l'application TTLs appropriée de vos clés mises en cache peut améliorer les performances et améliorer globalement l'expérience de jeu.

Étapes d'implémentation

- Utilisez les politiques de cycle de vie d'Amazon S3 pour transférer les objets vers des niveaux d'accès ou d'archivage peu fréquents et configurez des actions d'expiration afin de supprimer les objets inutiles en fonction des règles du cycle de vie.
- Activez le Time to Live (TTL) dans les tables DynamoDB pour supprimer automatiquement les éléments expirés sans consommer de débit d'écriture, en définissant l'horodatage d'expiration à l'époque Unix.
- Définissez TTLs des ElastiCache clés appropriées en fonction des taux de modification des données et de la tolérance au risque pour les données périmées, afin de faciliter l'actualisation des données mises en cache et d'améliorer l'expérience des joueurs.

Matériel et services

GAMESUS02 : Comment gérez-vous l'utilisation des ressources informatiques dans le backend de vos jeux ?

Les studios doivent développer une stratégie informatique qui utilise une combinaison de différents types de calcul, de services gérés et de plans d'économies pour optimiser votre utilisation. Vous devez également optimiser la façon dont les serveurs de jeu et les services principaux sont intégrés aux instances de calcul afin de réduire le nombre de ressources inutiles.

Bonnes pratiques

- GAMESUS02-BP01 Sélectionnez les services gérés pour les charges de travail de calcul appropriées
- GAMESUS02-BP02 Ajustez votre capacité de calcul et déployez les performances du GPU uniquement là où cela est nécessaire

GAMESUS02-BP01 Sélectionnez les services gérés pour les charges de travail de calcul appropriées

Concevez les services de backend de votre jeu de manière à utiliser des services gérés pour des charges de travail liées à des événements ou à des charges de trafic très variables. Les services gérés transfèrent la gestion de l'infrastructure vers plusieurs utilisateurs AWS et répartissent l'impact environnemental entre plusieurs utilisateurs en raison des plans de contrôle multi-locataires.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

AWS des services tels que AWS Lambda, AWS Fargate (Containers) et Amazon Gamelift (orchestration de serveurs de jeux) peuvent exécuter du code, des conteneurs ou orchestrer vos serveurs de jeux sans avoir à gérer l'infrastructure sous-jacente. Ces services évoluent automatiquement en fonction de la demande des joueurs et seules les ressources que vous consommez vous sont facturées. L'infrastructure sous-jacente étant gérée en votre nom, vous pouvez vous concentrer uniquement sur les exigences de vos jeux et de vos services principaux.

Vous pouvez utiliser AWS Lambda pour exécuter du code sans allouer ou gérer des serveurs. Lambda exécute votre code sur une infrastructure informatique à haute disponibilité et gère les ressources de calcul, notamment la maintenance des serveurs et des systèmes d'exploitation, le provisionnement des capacités, le dimensionnement automatique et la journalisation. Avec Lambda, vous devez fournir votre code dans l'un des environnements d'exécution de langage pris en charge par Lambda. Lambda est utile pour traiter les événements du jeu, l'authentification des joueurs, le traitement des achats en jeu et les demandes de matchmaking. Lambda évolue automatiquement en fonction du nombre d'événements et peut gérer des pics de trafic inattendus.

AWS Fargateest un moteur de calcul sans serveur pour les conteneurs qui fonctionne à la fois avec Amazon Elastic Container Service (ECS) et Amazon Elastic Kubernetes Service (EKS). AWS Fargate vous permet de vous concentrer facilement sur le développement de vos applications en simplifiant le provisionnement et la gestion des serveurs, en vous permettant de spécifier et de payer les ressources par application, et en améliorant la sécurité grâce à l'isolation des applications dès la conception. Fargate est idéal pour les services backend qui gèrent les profils des joueurs, la gestion des états et le matchmaking.

Amazon GameLift est un service géré permettant de déployer, d'exploiter et de dimensionner des serveurs de jeu dédiés aux jeux multijoueurs basés sur des sessions. Vous pouvez déployer votre premier serveur de jeu dans le cloud en quelques minutes, ce qui vous permet d'économiser jusqu'à des milliers d'heures d'ingénierie lors du développement logiciel initial et de réduire les risques techniques qui poussent souvent les développeurs à supprimer les fonctionnalités multijoueurs de leurs conceptions.

Étapes d’implémentation

- Utilisez-le AWS Lambda pour les charges de travail axées sur les événements, telles que le traitement des événements de jeu, l'authentification des joueurs, les achats en jeu et les demandes de matchmaking, en tirant parti de sa mise à l'échelle automatique et de sa gestion sans serveur.
- Déployez AWS Fargate avec ECS ou EKS pour les services principaux tels que les profils des joueurs, la gestion des états et le matchmaking, en supprimant la gestion des serveurs et en améliorant l'isolation des applications.
- Utilisez Amazon GameLift pour déployer et faire évoluer des serveurs de jeu dédiés aux jeux multijoueurs basés sur des sessions, réduisant ainsi le temps de développement et la complexité opérationnelle.

GAMESUS02-BP02 Ajustez votre capacité de calcul et déployez les performances du GPU uniquement là où cela est nécessaire

Concevez vos serveurs de jeu et votre backend pour utiliser efficacement les ressources informatiques. Le surprovisionnement des capacités de calcul peut entraîner des coûts inutiles et minimiser la quantité de ressources inactives ou sous-utilisées. Les instances GPU doivent être utilisées pour soutenir des efforts de développement spécifiques, tels que les reconstructions HLOD dans Unreal, ou si vos serveurs de jeu en ont besoin par conception. Cela réduit considérablement l'impact environnemental et les coûts de vos charges de travail.

Niveau d’exposition au risque si cette bonne pratique n’est pas respectée : élevé

Directives d’implémentation

Vous devez optimiser vos serveurs de jeu et vos services principaux pour utiliser plusieurs types d' EC2 instances et le moins d'instances nécessaires. Cela augmente le nombre d'instances disponibles pour répondre à vos besoins pendant le développement ou pour le lancement de vos jeux. Vous devez également faire correspondre le type d'instance à la charge de travail spécifique que vous déployez. Les instances optimisées pour le calcul prennent en charge un large éventail de cas d'utilisation, notamment les serveurs de jeux et les services principaux tels que le matchmaking. Les instances optimisées pour la mémoire sont conçues pour fournir des performances rapides aux charges de travail qui traitent de grands ensembles de données en mémoire. Utilisez les instances de GPU selon les besoins en termes de performances élevées, mais pas pour les tâches informatiques générales. Si possible, concevez vos services ou serveurs de jeux pour qu'ils s'exécutent sur ARM avec des instances AWS Graviton. Graviton est le type d'instance le plus performant et le plus économe en énergie disponible sur AWS. Elles offrent également des performances et des coûts améliorés par rapport aux types d'instances x86.

Utilisez-les Optimiseur de calcul AWSpour identifier les configurations de AWS ressources optimales, telles que les types d'instances Amazon Elastic Compute Cloud (EC2), les configurations de volume Amazon Elastic Block Store (EBS), la taille des tâches des services Amazon Elastic Container Service (ECS) activés, les licences logicielles commerciales AWS Fargate, les tailles de mémoire AWS Lambda fonctionnelle et les classes d'instances de base de données Amazon Relational Database Service (RDS), en utilisant le machine learning pour analyser les indicateurs d'utilisation historiques. Compute Optimizer fournit un ensemble APIs et une expérience de console pour réduire les coûts et améliorer les performances des charges de travail en recommandant les AWS ressources optimales pour vos charges de travail. AWS

Étapes d’implémentation

- Associez les ressources informatiques à des charges de travail spécifiques en utilisant des instances optimisées pour le calcul pour les serveurs de jeux, des instances optimisées pour la mémoire pour les grands ensembles de données et des instances GPU uniquement pour des tâches telles que les reconstructions HLOD ou les serveurs de jeu dépendants du GPU.
- Optimisez l'utilisation du calcul en déployant des instances AWS Graviton dans la mesure du possible pour une efficacité énergétique, de meilleures performances et des économies par rapport aux instances x86.
- Optimiseur de calcul AWS À utiliser pour analyser l'historique d'utilisation et recommander les configurations les plus efficaces pour EC2 les charges de travail AWS ECS et Amazon RDS afin de réduire les coûts et d'améliorer les performances. AWS Lambda

Ressources

Consultez les ressources suivantes pour en savoir plus sur nos meilleures pratiques en matière de développement durable.

- ONU : L'industrie du jeu met en lumière les menaces qui pèsent sur la planète
- Médium : Durabilité environnementale dans le développement de jeux : jouer de manière responsable pour un avenir plus vert

AWS Principaux services

- Amazon Aurora
- Amazon DynamoDB
- Amazon DocumentDB (compatible avec MongoDB)
- Amazon ElastiCache
- Amazon S3
- Classes de stockage Amazon S3
- Classe de stockage Amazon S3 Intelligent-Tiering
- Classe de stockage Amazon S3 Express One Zone
- Amazon Elastic Block Store
- AWS Lambda

- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service
- Amazon GameLift
- Optimiseur de calcul AWS

Conclusion

Les jeux sont conçus pour offrir des expériences de divertissement à un public mondial de joueurs et présentent des caractéristiques d'utilisation généralement imprévisibles et variables. Le Game Industry Lens décrit les types courants de scénarios qui constituent généralement une architecture de jeu et fournit un ensemble de questions et de bonnes pratiques à prendre en compte lors de la création et de l'exploitation de jeux dans le cloud. En appliquant ce framework à votre architecture de jeu, vous pouvez créer des jeux fiables, sécurisés, efficaces et économiques dans le cloud.

Collaborateurs

Les personnes suivantes ont contribué à ce document :

- Adam Hatfield, architecte de solutions senior, Amazon Web Services
- Brady Webb, responsable des comptes techniques, Amazon Web Services
- Bruce Ross — Architecte de solutions senior, responsable de Well-Architected Lens, Amazon Web Services
- Caleb Cecil, architecte de solutions associé, Amazon Web Services
- Carlos Perez, Cloud Optimization Success SA, Amazon Web Services
- Chase Herrington, responsable des comptes techniques ESL, Amazon Web Services.
- Chris Blackwell, architecte de solutions senior, Amazon Web Services
- Corey Ouderkirk, directeur technique principal des comptes, Amazon Web Services
- Derek Villavicencio, société de prototypage SA, Amazon Web Services
- Erik Ynigo Becerril, architecte de solutions senior, Amazon Web Services
- Grzegorz Ochmanski, architecte de solutions senior, Amazon Web Services
- Hadrian Baron, directeur technique principal des comptes, Amazon Web Services
- Ian Armbruster, responsable principal des solutions clients, Amazon Web Services
- Jed Obray, directeur technique principal des comptes, Amazon Web Services
- Khurram Khokhar, responsable technique principal des comptes, Amazon Web Services
- Kyle Somers, directeur principal de l'architecture des solutions, Amazon Web Services
- Madhuri Srinivasan, rédactrice technique senior, Well-Architected, Amazon Web Services
- Matthew Wygant, conseiller principal en matière de TPM, Well-Architected, Amazon Web Services
- Nataliya Godunok, Cloud Optimization Success SA, Amazon Web Services
- Nirav Doshi, architecte de solutions principal, Amazon Web Services
- Olivia Liddell, architecte de solutions, Amazon Web Services
- Randy James, directeur technique principal des comptes, Amazon Web Services
- Reou Ando, architecte de solutions de jeux, Amazon Web Services
- Richard Raseley, directeur technique principal des comptes, Amazon Web Services
- Sam Patzer, architecte de solutions senior, Amazon Web Services
- Scott Selinger, architecte de solutions senior, Amazon Web Services

- Sean Allen, architecte de solutions senior, Amazon Web Services
- Serge Poueme, architecte de solutions senior, Amazon Web Services
- Stewart Matzek, rédacteur technique senior, Well-Architected, Amazon Web Services
- Trenton Potgieter, architecte de solutions senior, Amazon Web AI/ML/Analytics Services

Révisions du document

Pour être informé des mises à jour de ce livre blanc, abonnez-vous au flux RSS.

Modification	Description	Date
Nouvelle version d'objectif	L'intégralité de l'objectif a été mise à jour avec de nouveaux conseils sur les meilleures pratiques.	9 décembre 2025
Publication initiale	Livre blanc publié pour la première fois.	19 novembre 2021

AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le AWS glossaire dans la Glossaire AWS référence.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.