

AWS Weißbuch

Linse für die Spieleindustrie



Linse für die Spieleindustrie: AWS Weißbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Zusammenfassung und Einführung	i
Verfügbarkeit von Objektiven	2
Designprinzipien	3
Szenarien	5
Game-Hosting für synchrones Gameplay in Echtzeit	5
Spieleserver-Prozesse	6
Sitzungsbasiertes Gameserver-Hosting mit serverlosem Backend	8
Multi-Regions- und Hybridarchitektur für Spiele mit niedriger Latenz	10
Backends für Spiele	12
Container-basierte Spiele-Backend-Architektur	12
Serverlose Backend-Architektur für Spiele	15
Entwicklung von Cloud-Spielen (CGD)	18
Entwicklung von Cloud-Spielen: CI/CD	19
Entwicklung von Cloud-Spielen: Workstations	21
Spieleanalytik-Pipeline	23
Definitionen	26
Spielsystem	27
Spieleserver	28
Spielclient	30
Messaging	31
Live-Spielbetrieb (Live Ops)	32
Operative Exzellenz	33
Designprinzipien	33
Live-Betrieb	34
GAMEOPS01-BP01 Verwenden Sie Spielziele und Kennzahlen zur Unternehmensleistung, um Ihre Live-Betriebsstrategie zu entwickeln	35
Struktur des Kontos	36
GAMEOPS02-BP01 Verwenden Sie eine Strategie für mehrere Konten, um verschiedene Spiele und Anwendungen in ihren eigenen Konten zu isolieren	36
GAMEOPS02-BP02 Organisieren Sie Infrastrukturressourcen mithilfe von Ressourcen- Tagging	41
Bereitstellungen von Spielen	41
GAMEOPS03-BP01 Validiere und teste deine bestehenden Kernspielsysteme und - infrastrukturen, bevor du sie in deinem Spiel wiederverwendest	42

GAMEOPS03-BP02 Führen Sie vor jeder Veröffentlichung (oder zumindest bei Hauptversionen) ein Performance-Engineering durch	43
GAMEOPS03-BP03 Testen Sie früh und häufig	44
GAMEOPS03-BP04 Verfolgen Sie eine Einsatzstrategie, die die Auswirkungen auf die Spieler minimiert	47
GAMEOPS03-BP05 Zur Unterstützung von Spitzenanforderungen ist eine vorab skalierbare Infrastruktur erforderlich	51
Überwachung Health Gesundheitszustands	54
GAMESOPS04-BP01 Instrumentieren Sie das Spiel, um Probleme zu erkennen und zu überwachen, die sich auf die Spieler auswirken	54
Lasttest	55
GAMEOPS05-BP01 Wählen Sie die richtige Phase, Architektur und das richtige Lasttesting-Framework, um Ihre Ziele zu erreichen	56
Optimierung im Laufe der Zeit	60
GAMEOPS06-BP01 Überwachen Sie wichtige Spielmetriken, um Spielertrends und -muster zu identifizieren, und nutzen Sie diese Informationen, um das Spiel zu verbessern	60
GAMEOPS06-BP02 Aktualisiere und passe den Lasttestansatz an, wenn sich das Spiel ändert	61
Ressourcen	63
Dokumentation und Blogs	63
Partnerlösungen	64
Whitepaper	65
Videos	65
Materialien für die Ausbildung	65
Sicherheit	66
Designprinzipien	67
Sicherheitsgrundlagen	67
GAMESEC01-BP01 Verwenden Sie Rollen und Verbundzugriff und nicht den Root-Benutzer des Kontos, um Aktionen in Ihrer Umgebung durchzuführen AWS	68
GAMESEC01-BP02 Wird verwendet, um schnell eine Umgebung mit mehreren Konten einzurichten auf AWS Control Tower AWS	69
GAMESEC01-BP03 Verwenden Sie Rollenrichtlinien mit den geringsten Rechten, die auf bestimmte Berufsfunktionen zugeschnitten sind	71
GAMESEC01-BP04 Verwenden Sie Rollen und Verbundzugriffsrichtlinien zusammen mit Zugriffsrichtlinien auf Kontoebene, um Zugriff auf Ihre Ressourcen zu gewähren AWS	72
GAMESEC01-BP05 Verwenden Sie einen zentralen Identitätsanbieter	73

Kontinuierliche Sicherheit	74
GAMESEC02-BP01 Verwenden Sie einsatzbereite Vorlagen für Standardsicherheitspraktiken	75
GAMESEC02-BP02 Verwenden Sie automatisierte Abhilfemaßnahmen, wenn ein Sicherheitsereignis eintritt	76
Identity and Access Management	78
GAMESEC03-BP01 Legen Sie fest, wie Sie den Zugriff von Spielern auf die Umgebung und die Ressourcen Ihres Spiels identifizieren und kontrollieren möchten	78
GAMESEC03-BP02 Authentifizieren Sie Anfragen, die an den Backend-Dienst Ihres Spiels gesendet werden	80
GAMESEC03-BP03 Verwende deinen Spiel-Backend-Service, um Spieleranfragen zur Teilnahme an einem Multiplayer-Spiel zu validieren	82
GAMESEC03-BP04 Setzen Sie strenge Sicherheitsrichtlinien für Spieler-Benutzerkonten durch, indem Sie ein sicheres Passwort verlangen	84
GAMESEC03-BP05 Bieten Sie Spielern die Möglichkeit, die Multi-Faktor-Authentifizierung (MFA) für ihre Konten einzurichten	84
Zugriffskontrolle	85
GAMESEC04-BP01 Beschränken Sie den Zugriff auf herunterladbare Inhalte auf autorisierte Kunden und Benutzer	86
GAMESEC04-BP02 Beschränken Sie den ursprünglichen Zugriff auf autorisierte Netzwerke zur Inhaltsbereitstellung () CDNs	89
GAMESEC04-BP03 Implementieren Sie geografische Beschränkungen, um unbefugten Zugriff einzuschränken	90
GAMESEC04-BP04 Beschränken Sie den Zugriff auf Inhalte mit DRM-Lösungen (Digital Rights Management)	91
Erkennung	92
GAMESEC05-BP01 Implementieren Sie eine umfassende Datenerfassungsstrategie zur Überwachung des Spielerverhaltens	93
GAMESEC05-BP02 Sammeln, speichern und analysieren Sie Nutzungsprotokolle von Spielern, um unangemessenes Verhalten zu erkennen	94
Schutz der Infrastruktur	95
GAMESEC06-BP01 Verwenden Sie Tools, um Bedrohungen für Ihre Infrastruktur zu erkennen und darauf zu reagieren	95
GAMESEC06-BP02 Verwenden Sie Tools für künstliche Intelligenz und maschinelles Lernen, um Aspekte Ihrer Infrastrukturschutzstrategie zu automatisieren	97

GAMESEC06-BP03 Nutzen Sie Erkenntnisse aus Protokollen auf Systemebene, um Ihre Strategie zum Schutz der Infrastruktur kontinuierlich zu verbessern	98
Vorfallreaktion	99
GAMESEC07-BP01 Implementieren Sie einen Plan zur Reaktion auf Vorfälle, um mit schlechten Akteuren und missbräuchlichem Verhalten umzugehen	100
GAMESEC07-BP02 Sperren Sie Konten, die mit schlechten Schauspielern in Verbindung stehen	101
Anwendungssicherheit	102
GAMESEC08-BP01 Wenden Sie Sicherheit in jeder Phase der CI/CD-Pipeline an	103
Automatisieren Sie die Sicherheit	104
GAMESEC09-BP01 Integrieren Sie Tools und Automatisierung, um die durchschnittliche Zeit für Sicherheitsüberprüfungen zu reduzieren	104
Modellierung von Bedrohungen	105
GAMESEC10-BP01 Legen Sie fest, wann und wie Sie die Bedrohungsmodellierung während des gesamten Lebenszyklus der Anwendungsentwicklung durchführen sollten	106
Ressourcen	107
Zuverlässigkeit	109
Designprinzipien	109
Grundlagen	110
Workload-Architektur	110
GAMEREL01-BP01 Verteilen Sie die Spieleinfrastruktur auf mehrere Availability Zones und Regionen, um die Ausfallsicherheit zu verbessern	110
Änderungsmanagement	113
GAMEREL02-BP01 Implementieren Sie eine Skalierungsstrategie, die den Status der Spielsitzungen aktiver Spieler berücksichtigt	113
GAMEREL02-BP02 Support die Verwendung mehrerer EC2 Instanztypen für dein Spiel	114
Fehlerverwaltung	115
GAMEREL03-BP01 Überwachen Sie Störungen der Spieleserver und nutzen Sie die Daten, um die Hosting-Architektur zu verbessern und so die Zuverlässigkeitsziele zu erreichen	116
GAMEREL03-BP02 Implementieren Sie eine lose Kopplung von Spielfunktionen, um Ausfälle mit minimalen Auswirkungen auf das Spielerlebnis zu beheben	117
GAMEREL03-BP03 Überwachen Sie Infrastrukturereignisse im Zeitverlauf, um die Auswirkungen auf das Spielerverhalten zu messen	120
Ressourcen	121
Leistungseffizienz	123
Designprinzipien	123

Auswahl der Architektur	124
GAMEPERF01-BP01 Evaluieren Sie die Ressourcenanforderungen und Skalierbarkeitsanforderungen für Spielleserver	125
GAMEPERF01-BP02 Berücksichtigen Sie den Betriebsaufwand für die Skalierung von Spielleservern	127
GAMEPERF01-BP03 Evaluieren Sie die Integration mit anderen AWS Diensten, Entwicklungsumgebungen, Ziel-CPU-Architekturen und Funktionen	128
Auswahl der Region	129
GAMEPERF02-BP01 Wähle eine Heimatregion in der Nähe deiner Spieler	130
GAMEPERF02-BP02 Entwickeln Sie einen Ansatz, der die Platzierung einer latenzempfindlichen Spieleinfrastruktur in der Nähe der Spieler unterstützt, um die Leistung zu verbessern	131
Iterative Entwicklung	134
GAMEPERF03-BP01 Verwenden Sie Amazon GameLift Anywhere und ein Test-Toolkit GameLift	134
GAMEPERF03-BP02 Testen Sie die Leistung und Skalierbarkeit von Spielleservern	136
GAMEPERF03-BP03 Optimieren Sie die Ressourcennutzung von Containern GameLift	137
Computer und Hardware	138
GAMEPERF04-BP01 Überwachen Sie die Spielserversprozesse, um Probleme zu erkennen	139
GAMEPERF04-BP02 Testen Sie die Leistung Ihres Gameservers mit simulierten und realen Gameplay-Szenarien	140
Auswahl berechnen	141
GAMEPERF05-BP01 Benchmarken Sie Ihre Spieleleistung auf mehreren Rechnertypen	141
GAMEPERF05-BP02 Verschieben Sie Rechenaufgaben in asynchrone Workflows non- latency-sensitive	143
Datenverwaltung	144
GAMEPERF06-BP01 Zentralisieren Sie die Erfassung und Speicherung von Protokollen	145
GAMEPERF06-BP02 Kategorisieren und speichern Sie Spieldaten auf der Grundlage von Zugriffsmustern	146
GAMEPERF06-BP03 Ermöglichen Sie effizientes Formatieren und Batching von Protokollen	147
GAMEPERF06-BP04 Implementieren Sie Richtlinien für die Protokollrotation und Aufbewahrung	147
GAMEPERF06-BP05 Verwenden Sie Überwachungs- und Visualisierungstools	148
Netzwerk und Bereitstellung von Inhalten	148

GAMEPERF07-BP01 Definieren Sie Schwellenwerte für die Netzwerklatenz für Ihr Spiel	149
GAMEPERF07-BP02 Führe für jeden Spielmodus und jede Spielhosting-Region einen separaten Matchmaking-Dienst aus	149
GAMEPERF07-BP03 Überwachen Sie regelmäßig die Matchmaking-Leistung	150
GAMEPERF07-BP04 Überwachen Sie regelmäßig die Netzwerkleistung	151
GAMEPERF07-BP05 Verwenden Sie die Netzwerkbeschleunigungstechnologie, um die Leistung im Internet zu verbessern	152
Prozess und Kultur	154
GAMEPERF08-BP01 Informieren Sie den Spieler und beziehen Sie ihn in Ihren Prozess ein	154
GAMEPERF08-BP02 Passen Sie die Lösungsauswahl an den Fähigkeiten und dem Fachwissen des Entwicklungsteams an	156
Ressourcen	156
Kostenoptimierung	159
Designprinzipien	160
Cloud Financial Management betreiben	161
Ausgabenerkennung und Nutzungsbewusstsein	161
GAMECOST01-BP01 Implementieren Sie die Kostenzuweisung pro Spieler, Spielfunktion und Umgebung	161
GAMECOST01-BP02 Entdecken Sie Optimierungsmöglichkeiten	163
Kostengünstige Ressourcen	164
GAMECOST02-BP01 Optimieren Sie die Kosten für die Datenübertragung über das Internet	165
GAMECOST02-BP02 Optimiere die Anzahl der Spielsitzungen, die auf jeder Spieleserverinstanz gehostet werden, um die Kosten zu optimieren	167
GAMECOST02-BP03 Wählen Sie die entsprechende Preisoption für Computeranwendungen aus, um die Kosten zu senken	168
Datenübertragungskosten	171
GAMECOST03-BP01 Wählen Sie den geeigneten Speichertyp für benutzergenerierte Inhalte, um die Kosten zu senken	171
GAMECOST03-BP02 Optimiere Datenbanken für Spiele-Backends	173
Verwaltung der Nachfrage- und Angebotsressourcen	175
Optimierung im Laufe der Zeit	175
Ressourcen	175
Nachhaltigkeit	177
Designprinzipien	177

Auswahl der Region	178
Ausrichtung am Bedarf	178
Software und Architektur	178
Datenverwaltung	178
GAMESUS01-BP01 Verwenden Sie Speichertechnologien, die den Mustern entsprechen, die an Benutzerinhalte, Abonnenteninformationen und Käufe im Spiel angepasst sind	179
GAMESUS01-BP02 Verwende Lebenszyklusrichtlinien oder TTL-Ablaufzeiten, um nicht benötigte Spiele-Benutzerdaten, Protokolldateien oder veraltete Inhalte zu löschen	181
Hardware und Services	184
GAMESUS02-BP01 Wählen Sie verwaltete Dienste für geeignete Rechenarbeitslasten aus	184
GAMESUS02-BP02 Passen Sie die Größe Ihrer Rechenleistung an und setzen Sie GPU- Leistung nur dort ein, wo sie benötigt wird	186
Ressourcen	187
Wichtige AWS Dienste	187
Schlussfolgerung	189
Mitwirkende	190
Dokumentversionen	192
AWS Glossar	193
.....	cxciv

Lens of Games Industry — AWS Well-Architected strukturiertes Framework

Datum der Veröffentlichung: 9. Dezember 2025 ([Dokumentversionen](#))

Das [AWS Well-Architected Framework](#) unterstützt Cloud-Architekten beim Aufbau einer sicheren, leistungsstarken, belastbaren und effizienten Infrastruktur für ihre Anwendungen und Workloads. Well-Architected basiert auf sechs Säulen — betriebliche Exzellenz, Sicherheit, Zuverlässigkeit, Leistungseffizienz, Kostenoptimierung und Nachhaltigkeit — und bietet Kunden und AWS Partnern einen konsistenten Ansatz, um Architekturen zu bewerten, Risiken zu minimieren und Designs zu implementieren, die einen Geschäftswert bieten.

In diesem Zusammenhang konzentrieren wir uns darauf, wie Sie Ihre Spiele-Workloads in der Cloud entwerfen, bereitstellen und gestalten können. AWS Cloud Wir definieren Komponenten, untersuchen gängige Workload-Szenarien und skizzieren Entwurfsprinzipien, die Sie bei der Anwendung des Well-Architected Framework unterstützen. Wir empfehlen Ihnen, mit dem Entwurf Ihrer Architektur zu beginnen, indem Sie die Best Practices und Fragen aus dem [AWS Well-Architected Framework-Whitepaper](#) berücksichtigen. Dieses Dokument enthält ergänzende Best Practices für Kunden aus der Spielebranche.

In diesem Dokument werden bewährte Verfahren beschrieben, mit denen die einzigartigen Merkmale der Entwicklung und des Betriebs von Spielen in der Cloud berücksichtigt werden sollen. Grundlage dafür sind unsere Erfahrungen in der Zusammenarbeit mit Entwicklern und Publishern der Spielebranche auf der ganzen Welt. Es bietet Anleitungen dazu, wie Sie Ihre Umgebung so gestalten und betreiben können, dass sie kostenoptimiert und skalierbar ist, um den Schwankungen der weltweiten Nachfrage gerecht zu werden. Dieser Leitfaden bietet auch Hinweise zur Sicherung Ihrer Spieleinfrastruktur und zur Optimierung der Leistung, um ein positives Spielerlebnis zu bieten.

Dieses Dokument richtet sich an Personen in technischen Funktionen, wie z. B. Chief Technology Officers (CTOs), technische Direktoren von Spielstudios, Architekten, Entwickler und Mitglieder des Betriebsteams. Nachdem Sie dieses Dokument gelesen haben, werden Sie sich mit AWS bewährten Methoden und Strategien vertraut machen, die Sie beim Entwerfen von Architekturen für Spiele anwenden sollten.

Verfügbarkeit von Objektiven

Maßgeschneiderte Objektivtwe erweitern die von bereitgestellten Best-Practice-Anleitungen AWS Well-Architected Tool. AWS WA Tool ermöglicht es Ihnen, Ihre eigenen [benutzerdefinierten Objektivtwe](#) zu erstellen oder Objektivtwe zu verwenden, die von anderen erstellt wurden und die mit Ihnen geteilt wurden.

Laden Sie das [Games Industry Lens AWS Well-Architected Tool aus dem öffentlichen AWS Well-Architected-Repository für benutzerdefinierte GitHub Objektivtwe](#) herunter und importieren Sie es, um mit der Überprüfung Ihres Spiele-Workloads zu beginnen.

Designprinzipien

Das AWS Well-Architected Framework identifiziert die folgenden allgemeinen Entwurfsprinzipien, um ein gutes Design in der Cloud für Spiele-Workloads zu ermöglichen:

- Verstehen Sie das Verhalten und die Nutzungsmuster der Spieler, um das Spiel weiterzuentwickeln und die Spieler zu schützen: Um Ihr Spiel kontinuierlich zu verbessern und das Spielerlebnis effektiv zu gestalten, ist es wichtig, einen Überblick darüber zu haben, wie die Spieler mit dem Spiel selbst und mit den anderen Spielern interagieren. Dies hilft Ihnen zu verstehen, wie Sie das Spiel verbessern, die Kosten kontrollieren und unautorisierte Nutzungsaktivitäten, die das Spielerlebnis gefährden, überwachen und darauf reagieren können.
- Verwenden Sie Technologien, die den Spielbetrieb vereinfachen und die Entwicklungsgeschwindigkeit erhöhen: Priorisieren Sie die Einführung von Technologien, die die Geschwindigkeit verbessern und den Betriebsaufwand für die Bereitstellung neuer Funktionen und Verbesserungen für Spieler reduzieren können. Spiele sind trefferorientiert und es gibt viele Optionen, die die Spieler in Betracht ziehen müssen. Daher ist es entscheidend für den Erfolg eines Spiels, schnell voranzukommen und sich an Veränderungen anzupassen. Überlegen Sie, ob Sie mit dem Betrieb Ihrer eigenen Software vertraut sind oder ob Sie lieber einen vergleichbaren Managed Service von AWS Partnern oder AWS beidem nutzen möchten.
- Optimieren Sie Ihre Architektur, um Kennzahlen zu verbessern, die das tatsächliche Spielerlebnis widerspiegeln: Denken Sie bei der Anpassung und Weiterentwicklung Ihrer Architektur im Laufe der Zeit darüber nach, wie sich diese Verbesserungen und Änderungen auf das Spielerlebnis auswirken werden. Die Workloads in Spielen sollten in der Lage sein, den Auswirkungen von Ausfällen standzuhalten und diese so gering wie möglich zu halten, um weitverbreitete Spielunterbrechungen zu verhindern. Spielfunktionen und Systeme, die nicht entscheidend voneinander abhängig sind, sollten entkoppelt werden, um die Auswirkungen von Ausfällen zu reduzieren und Probleme zu isolieren, die sich auf die Spieler auswirken.
- Entwerfen Sie die Infrastruktur so, dass sie auch in Spitzenzeiten gleichzeitig spielt, und skalieren Sie sie dynamisch nach Bedarf: Die Infrastruktur sollte so konzipiert sein, dass sie der Nachfrage der Spieler gerecht wird. Metriken wie die Parallelität der Spielersitzungen und die Anzahl der Anmeldungen können verwendet werden, um präventiv zu skalieren, bevor die Systeme überlastet werden. Messwerte zur reaktiven Systemauslastung, wie z. B. CPU- und Speicherverbrauch, können zur Skalierung verwendet werden, wenn Systeme überlastet sind. Durch die dynamische Skalierung deiner Infrastruktur kannst du die Betriebskosten deines Spiels senken.

- Implementieren Sie Runbooks, um den Spielbetrieb zu verbessern: Operationelle Runbooks sollten verwendet werden, um wiederkehrende Aufgaben im Spielbetrieb einheitlich zu verwalten. Runbooks sollten für allgemeine Arbeitsabläufe im Spielbetrieb existieren, z. B. für die Untersuchung und Beantwortung von Spielermeldungen, die Verwaltung von Infrastrukturmaßnahmen vor der Skalierung zur Vorbereitung auf erwartete Großveranstaltungen wie den Start neuer Saisons und die Veröffentlichung von Spielinhalten sowie für typische Wartungsarbeiten am Spiel.

Szenarien

In diesem Abschnitt behandeln wir verschiedene Szenarien, die in einer Spielarchitektur üblich sind. Jedes Szenario enthält die gemeinsamen Merkmale, die dem Design zugrunde liegen, und ein Beispiel für ein Referenzarchitekturdiagramm.

Szenarien

- [Spiele-Hosting für synchrones Gameplay in Echtzeit](#)
- [Backends für Spiele](#)
- [Serverlose Backend-Architektur für Spiele](#)
- [Entwicklung von Cloud-Spielen \(CGD\)](#)
- [Spieleanalytik-Pipeline](#)

Spiele-Hosting für synchrones Gameplay in Echtzeit

Das synchrone Gameplay in Echtzeit ermöglicht es zwei oder mehr Spielern, gleichzeitig an einem Spiel teilzunehmen und zu interagieren, wobei der Status des Spiels zwischen den verbundenen Spielern geteilt wird, um ein möglichst echtzeitnahes Spielerlebnis zu schaffen. Beispiele für synchrone Spiele sind Ego-Shooter, MMOG (Massively Multiplayer Online Games), Sport- und Actionspiele oder Online-Spiele, bei denen zwei oder mehr Spieler miteinander verbunden sein müssen, um das Spielerlebnis nahezu in Echtzeit teilen zu können.

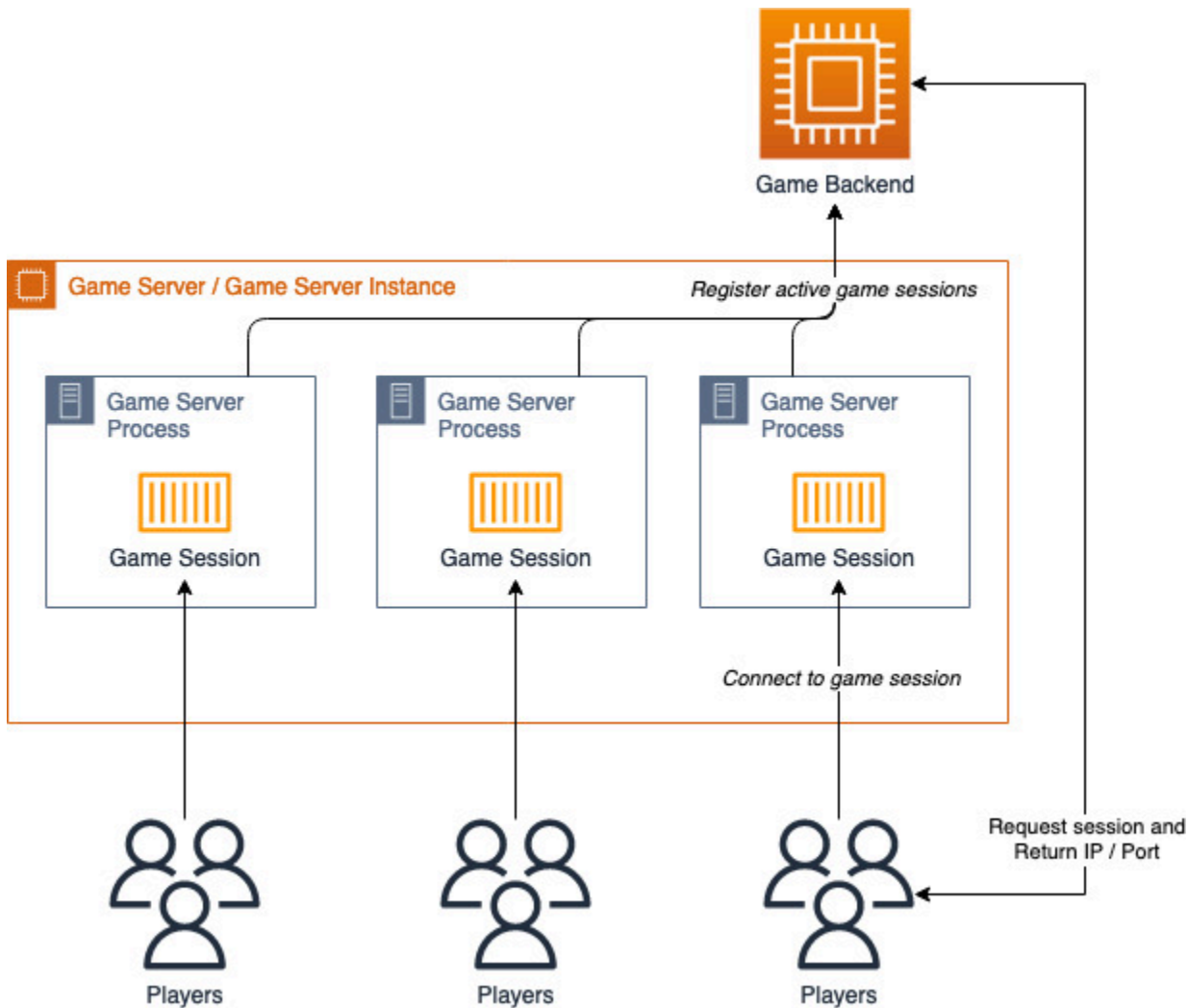
Zu den Merkmalen synchroner Echtzeit-Gameplay-Architekturen gehören:

- Einige Spiele können als Spielsitzungen über Spieleserverprozesse gehostet werden, die auf dedizierten Servern ausgeführt werden. Einige Spiele verwenden möglicherweise P2P-ähnliche Architekturen, die leichtere Session Traversal Utilities for NAT (STUN) oder Traversal Using Relays around NAT (TURN) -Server verwenden. Unabhängig vom verwendeten Servertyp werden Spieleserver in mehreren Rechenzentren und weltweit gehostet. AWS-Regionen
- Spielclients können an einer Spielsitzung teilnehmen, indem sie entweder ein Spiel von einem zentralen Matchmaking-Dienst anfordern, der im Backend-System des Spiels gehostet wird, oder indem sie ein Spiel aus einer vordefinierten Liste verfügbarer Spieleserver auswählen. Der Spielclient verfügt über eine IP-Adresse und einen Port, zu dem eine Verbindung hergestellt werden kann.

- Viele synchrone Spiele sind latenzempfindlich, wie Ego-Shooter und Massive-Multiplayer-Online-Spiele. Sie werden wahrscheinlich Algorithmen wie Rücklauf und Zeitdilatation enthalten, um Lag-Effekte zu minimieren, können aber auch eine vordefinierte Latenztoleranz haben, die sorgfältig gemessen und optimiert wird, um das Lag-Erlebnis zu reduzieren, das manchmal bei Spielern in Situationen mit hoher Latenz auftreten kann. Diese Latenzinformationen werden ermittelt, indem Spielclients so konfiguriert werden, dass sie den verfügbaren Spieleserver anpingen, AWS-Regionen um Messwerte wie Latenz, Netzwerkjitter und andere wichtige Messwerte für das Spielerlebnis zu erfassen. Diese Messwerte werden an einen zentralen Dienst zur Erfassung von Metriken im Backend-System des Spiels gesendet, sodass Live-Betriebsstreams den Zustand des Spiels überwachen können. Während des Matchmaking-Prozesses geben Spielclients ihre aktuellen Latenzdaten als einen der Anforderungsparameter an, wenn sie ein Spiel anfordern, und der Matchmaking-Dienst kann diese Latenzdaten als eine der Variablen verwenden, wenn er einen Spieleserver auswählt, der den Spieler hostet.
- In der Regel wird das Gameplay über eine Mischung von Protokollen abgewickelt (z. B. Spieleserver, die schnelleres UDP-basiertes Messaging mit Spielersuche, Authentifizierung und anderem Client-Server-Verkehr über HTTPS verwenden).
- Spieleserver verwenden Algorithmen und Designs, um den Client-Server-Verkehr wie Transformationsstreaming, Deltas und Datenkomprimierung zu minimieren.
- Spieleserver sind häufig das Ziel bössartiger Aktivitäten und sollten mit einer DDoS-Schutzlösung wie dieser geschützt werden. AWS Shield Advanced

Prozesse auf Spieleservern

Das folgende Diagramm zeigt eine typische Spieleserver-Architektur. Es beschreibt die logische Beziehung zwischen einer Spieleserverinstanz und den Spieleserverprozessen, die Spielsitzungen hosten.



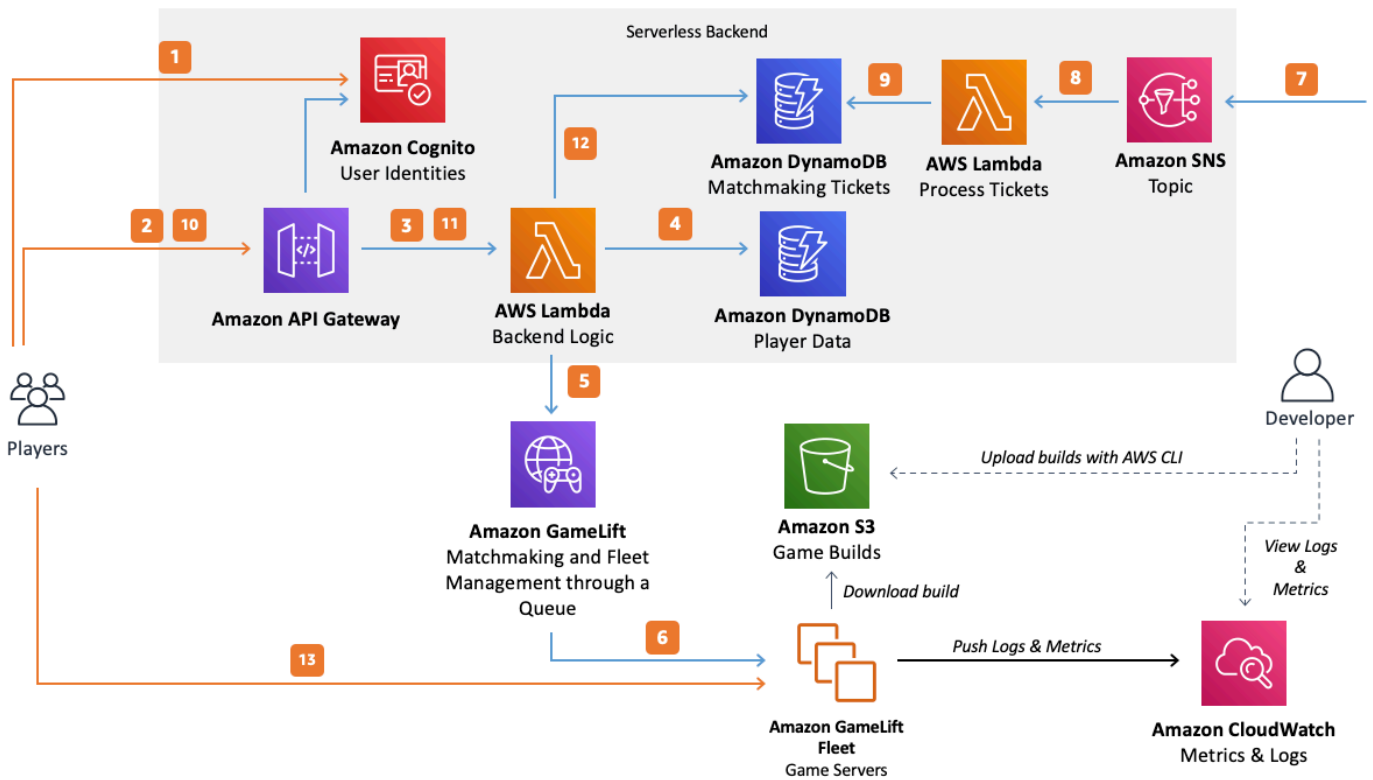
Logische Spieleserver-Architektur

- Eine EC2 Amazon-Instance wird als Spieleserver verwendet, auch Gameserver-Instance genannt. Der Spieleserver hostet einen oder mehrere Gameserver-Prozesse, und auf jedem läuft eine Kopie Ihres Gameserver-Builds. In der Regel werden mehrere Spieleserverprozesse auf einer Spieleserverinstance ausgeführt, um Rechenressourcen effizient zu nutzen und Kosten zu senken. Wenn eine Spielsitzung aktiv und bereit ist, Spilersitzungen zu veranstalten, wird ihr Status mit dem Spiel-Backend (normalerweise ein Matchmaking-Dienst) aktualisiert, sodass sie zum Hosten von Spielern verwendet werden kann.
- Das Spiel-Backend kann dem Spieler die IP-Adresse und den Serverport, der eine Spielsitzung hostet, zur Verfügung stellen, sodass er eine Verbindung zum Spielen herstellen kann.

Sitzungsbasiertes Gameserver-Hosting mit serverlosem Backend

Berücksichtigen Sie bei der Entwicklung einer Architektur für Ihr Spiel die Funktionen und Fähigkeiten, die Sie benötigen, und den Umfang des betrieblichen Verwaltungsaufwands, den Sie bereit sind, zu übernehmen. Um das beste Gleichgewicht zwischen einfacher Bedienung und Flexibilität zu erreichen, können Sie Ihr Spiel mithilfe von Managed Services von Cloud-Anbietern entwickeln. Mit Managed Services haben Sie die Kontrolle über die Entwicklung und Anpassung Ihrer eigenen Spielfunktionen und reduzieren gleichzeitig Ihren Aufwand für die Bereitstellung und Verwaltung der Infrastruktur.

Das Hosten eines sitzungsbasierten Multiplayer-Spiels erfordert eine Serverinfrastruktur zum Hosten der Spielserverprozesse sowie ein skalierbares Backend für Spiellersuche und Sitzungsmanagement. Die folgende Referenzarchitektur zeigt, wie Amazon GameLift Managed Hosting und ein serverloses Backend zur Verwaltung Ihrer sitzungsbasierten Spiele verwendet werden können.



Amazon GameLift verwaltetes Hosting für sitzungsbasierte Spiele

Das Diagramm beschreibt den Prozess, Spieler für Spiele zu gewinnen, die auf GameLift verwaltetem Spiele-Hosting laufen. Es umfasst die folgenden Schritte:

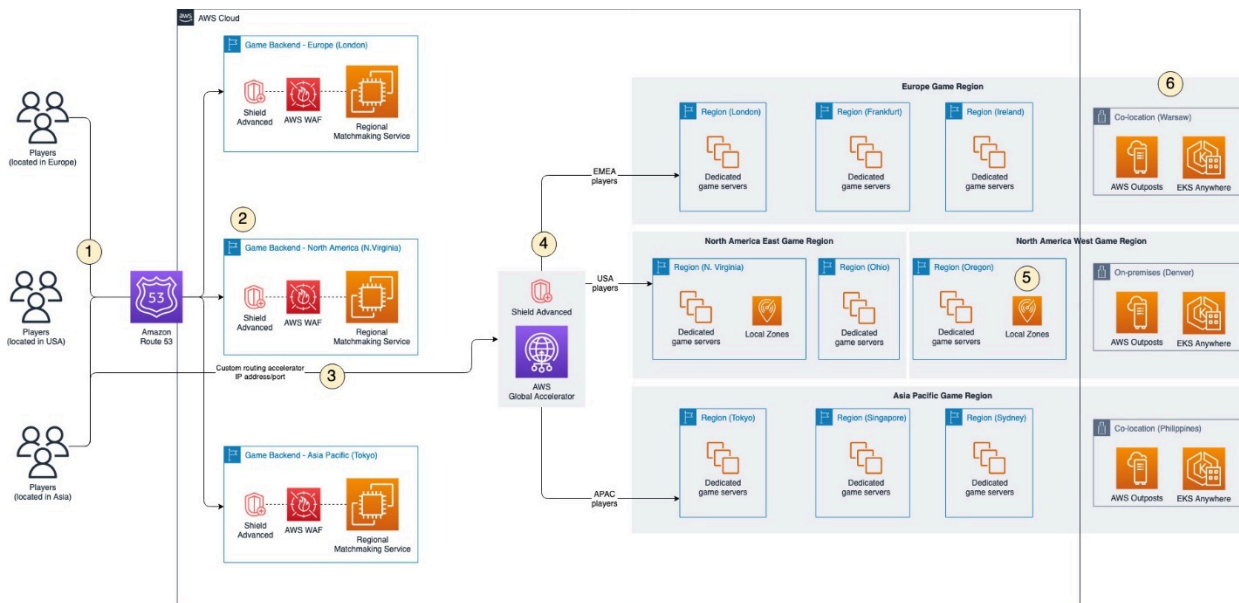
1. Der Spielclient fordert eine Amazon Cognito Cognito-Identität aus einem Amazon Cognito Cognito-Identitätspool an. Dieser kann optional mit externen Identitätsanbietern verbunden werden.
2. Der Spielclient erhält temporäre Zugangsdaten und fordert eine Spielsitzung über ein Amazon API Gateway an, indem er die Anfrage mit den Amazon Cognito Cognito-Anmeldeinformationen signiert.
3. API Gateway ruft eine AWS Lambda Funktion auf.
4. Die Lambda-Funktion fordert Spielerdaten aus einer Amazon DynamoDB-Tabelle an. Die Amazon Cognito Cognito-Identität wird verwendet, um sicher die richtigen Spielerdaten anzufordern, da die authentifizierte Identität in den Anforderungskontextdaten bereitgestellt wird.
5. Die Lambda-Funktion verwendet die richtigen Spielerdaten für zusätzliche Informationen (wie den Skill-Level des Spielers) und fordert über GameLift FlexMatch Matchmaking ein Match an. Sie können eine FlexMatch Matchmaking-Konfiguration mit JSON-basierten Konfigurationsdokumenten definieren. Der Spielclient kann Latenzmetriken generieren, indem er Serverendpunkte in verschiedenen Regionen anpingt, und die Latenzdaten können verwendet werden, um latenzbasiertes Matchmaking zu unterstützen.
6. Nach FlexMatch Matches mit einer geeigneten Gruppe von Spielern mit geeigneter Latenz in einer Region wird über eine Warteschlange eine Platzierung für eine Spielsitzung angefordert. GameLift Die Warteschlange enthält Flotten mit einem oder mehreren registrierten Standorten in der Region.
7. Wenn die Sitzung an einem der Standorte der Flotte stattfindet, wird eine Ereignisbenachrichtigung an ein Amazon SNS SNS-Thema gesendet.
8. Eine Lambda-Funktion empfängt das Amazon SNS SNS-Ereignis und verarbeitet es.
9. Wenn es sich bei der Amazon SNS-Nachricht um ein MatchmakingSucceeded Ereignis handelt, schreibt die Lambda-Funktion das Ergebnis mit dem Server-Port und der IP-Adresse in DynamoDB. Ein time-to-live (TTL) -Wert wird verwendet, um sicherzustellen, dass Matchmaking-Tickets aus DynamoDB gelöscht werden, wenn sie nicht mehr benötigt werden.
- 10Der Spielclient sendet eine signierte Anfrage an API Gateway, um den Status des Matchmaking-Tickets in einem bestimmten Intervall zu überprüfen.
- 11API Gateway ruft eine Lambda-Funktion auf, die den Status des Matchmaking-Tickets überprüft.
- 12Die Lambda-Funktion überprüft DynamoDB, um festzustellen, ob das Ticket erfolgreich war. Wenn dies erfolgreich war, sendet die Lambda-Funktion die IP-Adresse, den Port und die Player-Sitzungs-ID zurück an den Client. Wenn das Ticket fehlgeschlagen ist, sendet die Lambda-Funktion eine Antwort, in der erklärt wird, dass das Spiel nicht bereit ist.
- 13Der Spielclient stellt über TCP oder UDP eine Verbindung zum Spieleserver her, indem er den Port und die IP-Adresse verwendet, die vom Backend bereitgestellt werden. Es sendet die

Spieler Sitzungs-ID an den Spieleserver, und der Spieleserver validiert sie mithilfe des Amazon GameLift Server SDK.

Alternativ können Sie die vorherige Architektur ändern, um API Gateway WebSockets mit Amazon zu verwenden GameLift. Bei diesem Ansatz erfolgt die Kommunikation zwischen dem Spielclient und Ihrem Spiel-Backend-Service mithilfe einer [WebSocketbasierten Implementierung](#). Diese Implementierung kann verwendet werden, sodass die Lambda-Funktion des Spiele-Backends eine serverseitige Nachricht an den Spielclient über ein initiiert, WebSocket anstatt ein Abfragemodell zu implementieren.

Multi-Regions- und Hybridarchitektur für Spiele mit niedriger Latenz

In diesem Abschnitt wird eine multiregionale und hybride Architektur für Spiele mit niedriger Latenz beschrieben.



Verringerung der Latenz durch Netzwerkbeschleunigung und weltweit eingesetzte Spieleserver

1. Spieler in einem weltweit verfügbaren Spiel können von überall her kommen. Wenn ein Spieler eine Spielsitzung oder ein Spiel anfordert, sendet sein Spielclient eine Anfrage an den Spiel-Backend-Service, der bei Amazon Route 53 registriert ist. Das latenzbasierte Routing von Route 53 kann verwendet werden, um den Spieler zum nächstgelegenen verfügbaren Spiel-Backend weiterzuleiten.
2. Das Spiel-Backend wird in mehreren AWS-Regionen Populationen eingesetzt, die den Spielern am nächsten sind. Jedes Spiel-Backend umfasst einen regionalen Matchmaking-Service, der eine

- Spielsitzung aus allen Spielregionen findet. Obwohl die Matchmaking-Anfrage eines Spielers von einem regionalen Matchmaking-Dienst in seiner Nähe bearbeitet wird, kann der Matchmaking-Dienst einen Spieler bei Bedarf zu einer Spielsitzung in einer anderen Spielregion weiterleiten. Diese Aktion verbessert die Widerstandsfähigkeit und Leistung. Darüber hinaus verwendet jeder Back-End-Dienst für Spiele Layer-7-Webfilterung AWS WAF und AWS Shield Advanced Bot-Steuerung sowie Schutz vor Distributions-Denial-of-Service (S) -Angriffen und bietet Schutz vor Distributions-Denial-of-Service (S) -Angriffen. DDoS Sie haben viele Optionen, wie Sie Ihren Spiel-Backend-Service aufbauen möchten, z. B. serverlos, in Containern, EC2 Instanzen oder beim Hosten des Spiel-Back-End-Dienstes in Ihren eigenen Rechenzentren.
3. Um das Spielerlebnis zu verbessern, indem Netzwerklatenz und Jitter reduziert werden, wird mithilfe von AWS Global Accelerator ein benutzerdefinierter Routing-Beschleuniger bereitgestellt, der die Weiterleitung des Datenverkehrs vom Spielclient zum Spieleserver über das globale Netzwerk automatisch optimiert. AWS Sie konfigurieren den [benutzerdefinierten Routing-Beschleuniger](#) so, dass er die Global Accelerator-Listener-Ports dem Instanz-Port Ihres Spieleservers zuordnet. EC2 Der Spielclient stellt eine Verbindung mit der Global Accelerator-IP und dem Port her, der als Proxy dient und die Spieler deterministisch an die richtige Spielservers-IP und den richtigen Port weiterleitet, der die Spielsitzung hostet.
 4. Ihr Spiel umfasst spielerfreundliche logische Spielregionen, die eine Sammlung von Spieleserver-Hosting-Standorten darstellen, die geografisch nahe beieinander liegen, z. B. Nordamerika oder Asien-Pazifik. Um die Latenz zu reduzieren und die geografische Abdeckung zu erhöhen, kann eine Kombination verschiedener Hosting-Lösungen für Spieleserver verwendet werden, um das Spielerlebnis zu verbessern. Priorisieren Sie die Nutzung von, AWS-Regionen wo immer dies möglich ist, da diese Standorte über alle Funktionen verfügen und den größten Kapazitätsbedarf aufweisen.
 5. Verwenden Sie AWS Local Zones, um Spieleserver an geografischen Standorten unterversorgter Spieler zu hosten, an denen Sie nicht über Hosting-Einrichtungen verfügen oder eine nicht verfügbar AWS-Region ist.
 6. Stellen Sie AWS Outposts die Lösung in Ihren vorhandenen lokalen Rechenzentren und Colocation-Anbietern bereit, um mithilfe vollständig verwalteter Racks und Rack-Server eine nahtlose Steuerungsebene und Verwaltungserfahrung an jedem Bereitstellungsstandort zu schaffen. AWS Outposts sind auch von Vorteil, wenn Sie in Ihrer lokalen Umgebung nicht über vorhandene Serverkapazitäten verfügen. Wenn Sie jedoch eine Hybridimplementierung mit Software wünschen, die auf Ihrer eigenen vorhandenen Serverinfrastruktur ausgeführt wird, sollten Sie Amazon EKS Anywhere verwenden, mit dem Sie Kubernetes-Cluster auf Ihrer eigenen Infrastruktur mit Konnektivität zurück zu Amazon EKS erstellen und ausführen können. Dies bietet eine konsistente Konsolenansicht Ihrer Kubernetes-Cluster vor Ort.

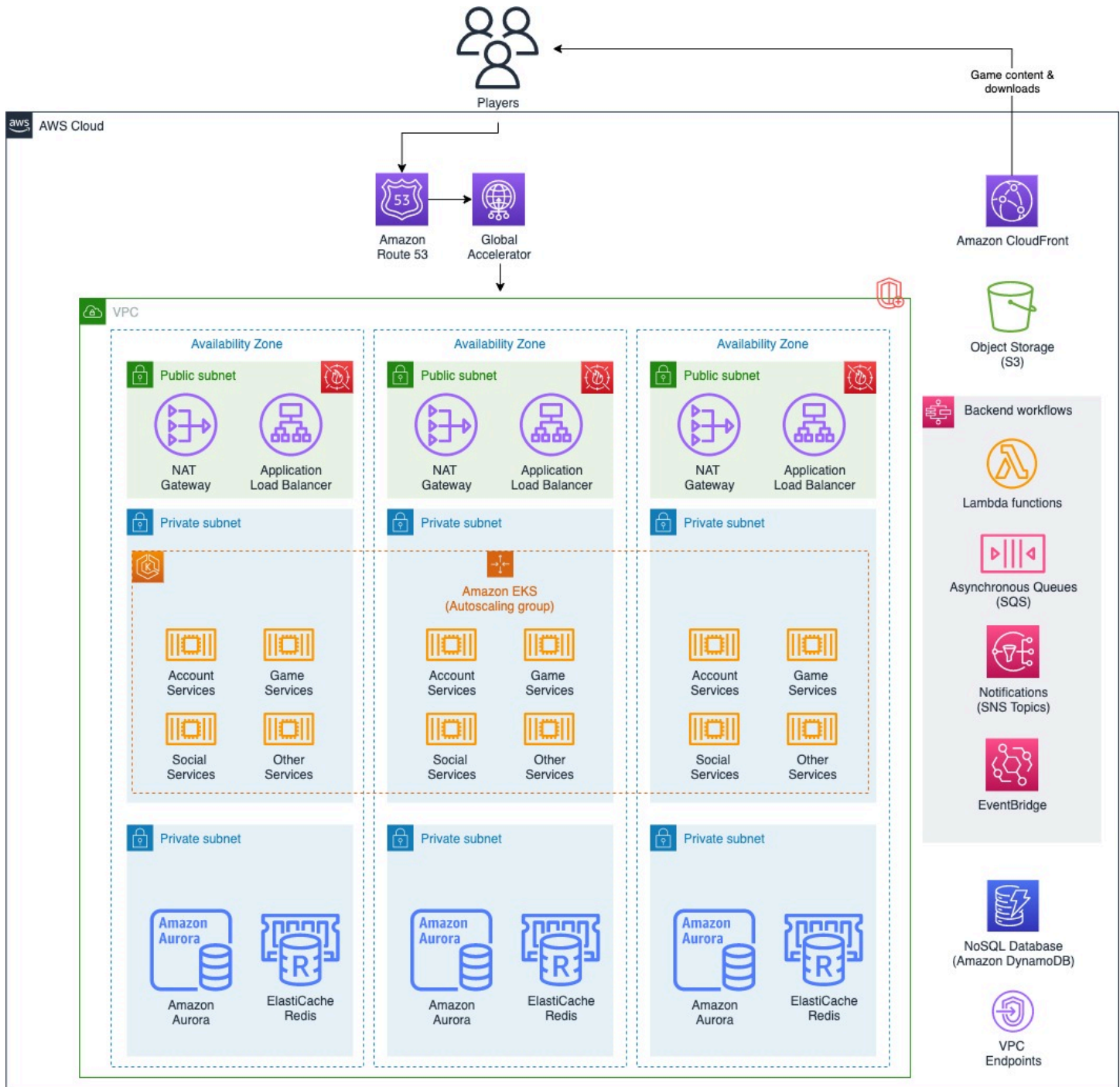
Backends für Spiele

Spiele-Backends werden verwendet, um das Spiel und den Spielerstatus zu verwalten sowie soziale Funktionen und Funktionen auf Systemebene in das Spiel zu integrieren, die das Spielerlebnis unterstützen. Die Verwaltung von Spielerprofilen, die Aufbewahrung von Gegenständen und Inventar sowie Statistiken und Bestenlisten sind Beispiele für Dienste, die in Spiel-Backends gehostet werden.

Spiele-Backends werden in der Regel als REST erstellt, auf APIs die Clients über HTTPS zugreifen. Es sind jedoch auch andere Ansätze üblich, z. B. WebSockets die Bereitstellung bidirektionaler Kanäle für Anwendungsfälle wie Client-Benachrichtigungen für Chat und Präsenz im Spiel. Spiele-Backends können mithilfe einer Vielzahl unterschiedlicher Bereitstellungsarchitekturen bereitgestellt werden, einschließlich der Verwendung von Instances, Containern oder einer serverlosen Architektur.

Container-basierte Backend-Architektur für Spiele

In diesem Abschnitt wird eine containerbasierte Backend-Architektur für Spiele beschrieben.



Hosten eines Spiele-Backends mithilfe von Containern

- Die Spieler greifen über eine Game-Client-Software auf das Spiel zu, die ihnen über ein Spielesystem, eine digitale Storefront oder einen direkten Download von einem Content Delivery Network (CDN) wie Amazon zur Verfügung gestellt werden kann. CloudFront CDNs bietet Caching an Edge-Standorten, um die Leistung für Benutzer beim Herunterladen von Inhalten zu

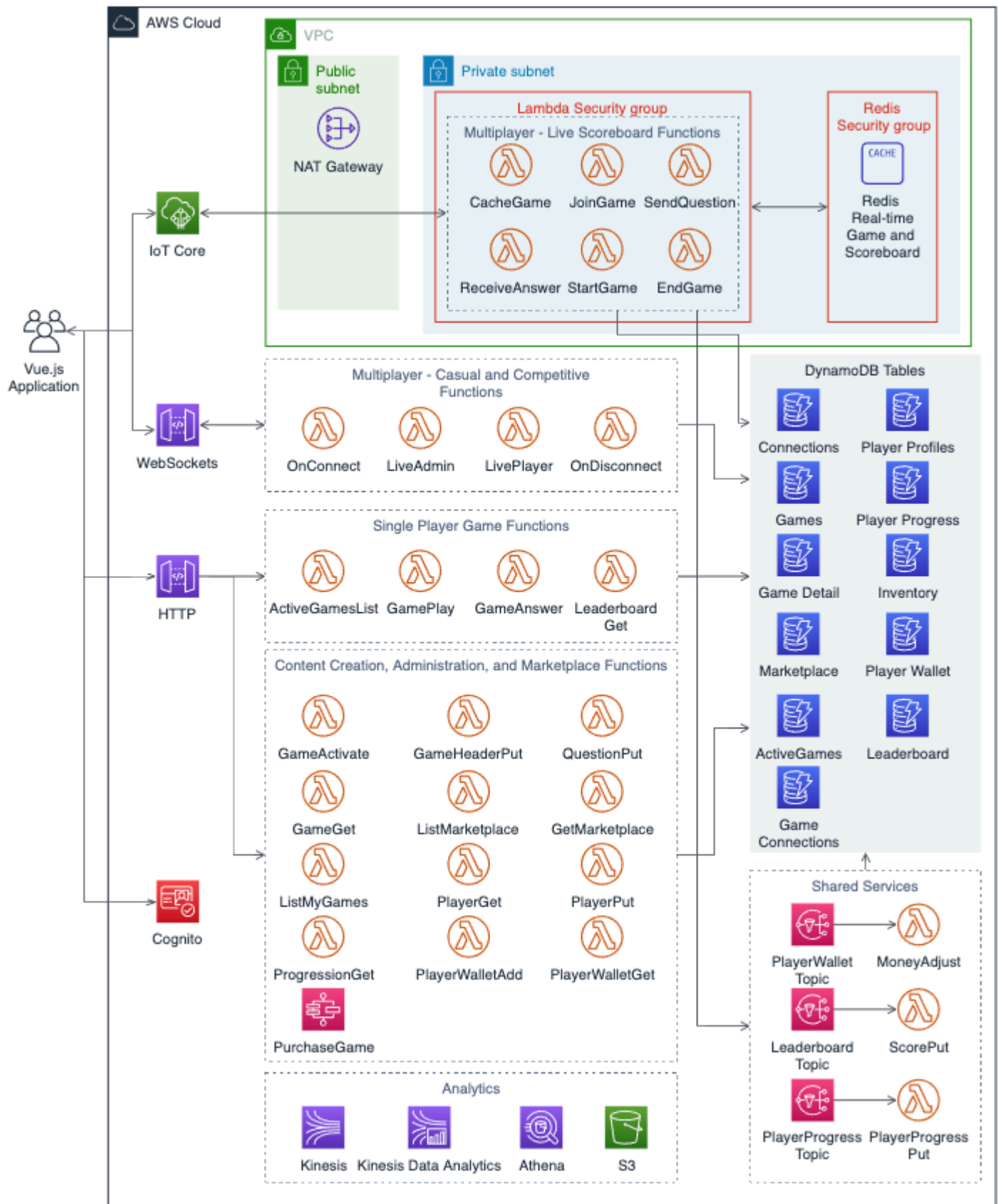
beschleunigen. CloudFront Kann beispielsweise verwendet werden, um die Game-Client-Software sowie die Spielressourcen und andere Inhalte an Ihre Spieler zu verteilen.

- AWS Global Accelerator bietet eine Beschleunigung des Datenverkehrs und anpassbare Steuerelemente für die Weiterleitung des Datenverkehrs von Spieler-Spielclients zu Ihren Load Balancern sowie für die regionenübergreifende Weiterleitung von Datenverkehr für Multi-Regions- und Failover-Zwecke. Die benutzerdefinierten Domainnamen für Ihr Game-Backend REST APIs sind in Amazon Route 53 so konfiguriert, dass der Datenverkehr an Global Accelerator-Endpunkte weitergeleitet wird. Nicht im Diagramm dargestellt, AWS Shield Advanced kann zusätzliche DDoS-Abwehr für Ihren Accelerator und Ihr Spiel-Backend bieten.
- NAT Gateway und Application Load Balancer werden in öffentlichen Subnetzen in jeder der Availability Zones bereitgestellt, die vom Spiel-Backend verwendet werden, um eine hohe Verfügbarkeit in der Region zu gewährleisten. AWS WAF wird auf dem Application Load Balancer bereitgestellt, um den Layer-7-Webdatenverkehr zu filtern.
- Das Spiel-Backend wird als Sammlung einzelner containerbasierter Microservices gehostet, die in einem Amazon EKS-Cluster in privaten Subnetzen bereitgestellt werden, die aus Gründen der Ausfallsicherheit über Availability Zones verteilt sind. Die automatische Skalierung passt die Kapazität der Dienste und Clusterknoten dynamisch an die Ressourcennutzung an, die in der Regel mit der Nachfrage der Spieler korreliert. Während Cluster Autoscaler die Anzahl der Knoten im Cluster automatisch anpasst, skaliert der Horizontal Pod Autoscaler automatisch die im Cluster bereitgestellten Pods.
- Spiel- und Spielerdaten werden in Backend-Datenbanken und Caches gespeichert, die in privaten Subnetzen über Availability Zones verteilt werden, wobei die Replikation zwischen Primär- und Replikatknoten erfolgt. Amazon Aurora ist eine beliebte Wahl für Anwendungsfälle wie Spielerprofile, Berechtigungen und In-Game-Käufe, die komplexere Abfrageanforderungen haben können und von den relationalen Datenmodellierungsfunktionen von MySQL und PostgreSQL profitieren können. Amazon ElastiCache ist nützlich, um leistungsstarke Bestenlisten und pub/sub Nachrichten zu erstellen und häufig abgerufene Daten zwischenspeichern, um die Latenz und die Belastung von Datenbanken zu reduzieren. Amazon DynamoDB ist ein vollständig verwalteter NoSQL-Datenspeicher, der sich ideal für unvorhersehbare Zugriffsmuster eignet und für Anwendungsfälle wie Spieler- und Spielstatusdaten, Sitzungsdaten, Inventar- und Artikelspeicher oder Anwendungsfälle, in denen Sie eine globale Datenbank mit minimalem Overhead benötigen, auf praktisch unbegrenzten Durchsatz skalieren kann.
- Asynchrone Verarbeitungsworkflows sollten verwendet werden, um Arbeiten auszuführen, die im Hintergrund ausgeführt werden können, z. B. das Aktualisieren von Bestenlisten oder das Senden von Freundschaftsanfragen. Konfigurieren Sie Ihr Spiele-Backend so, dass diese Art von Arbeit in

die Amazon SQS SQS-Warteschlangen verschoben wird, um sie an das Wachstum Ihres Spiels anzupassen, oder ziehen Sie in Betracht, Amazon SNS SNS-Themen zu verwenden, um die Arbeit parallel Parallelverarbeitung auf viele Warteschlangen für Verbraucheranwendungen zu verteilen. Verwenden Sie AWS Lambda-Funktionen, um die Verarbeitung ereignisgesteuert durchzuführen, um Ihre Kosten für die Recheninfrastruktur und den Verwaltungsaufwand zu reduzieren. Bei Workflows, die langlebig sind oder eine Aufgabenkoordination mit mehreren Schritten erfordern, sollten Sie erwägen, den gesamten Workflow mithilfe von zu orchestrieren. AWS Step Functions Amazon EventBridge kann verwendet werden, um Funktionen zu initiieren, um auf AWS Service- und benutzerdefinierte Anwendungsereignisse zu reagieren.

Serverlose Backend-Architektur für Spiele

Viele Spieleentwickler möchten die Infrastruktur nicht verwalten und ziehen es stattdessen vor, ihre Spiele mithilfe von Technologien zu entwickeln, die es ihnen ermöglichen, sich auf Software zu konzentrieren. In diesem Szenario wird eine serverlose Architektur empfohlen, da Sie damit Funktionen schneller und mit weniger Betriebsaufwand erstellen und veröffentlichen können. Serverlose Architekturen werden mithilfe von Cloud-Diensten entworfen, die je nach Bedarf dynamisch skaliert werden können, ohne dass Server eingerichtet, verwaltet und skaliert werden müssen. Die folgende Referenzarchitektur veranschaulicht, wie ein Spiel mithilfe einer serverlosen Architektur erstellt wird.



Serverlose Backend-Referenzarchitektur für Spiele

Diese Referenzarchitektur veranschaulicht ein webbasiertes Quizspiel, das Einzelspieler- und Mehrspielerfunktionen bietet.

- **Spielerauthentifizierung:** Spieler authentifizieren sich mit Amazon Cognito, das eine sichere Authentifizierung mit einem Benutzerverzeichnis für die Verwaltung der Spieleridentität bietet.
- **Spielelogik als serverlose Funktionen:** Spielfunktionen und Backend-Geschäftslogik werden als AWS Lambda Funktionen ausgeführt, die als Reaktion auf Ereignisse ausgelöst werden. Dadurch werden die Kosten niedrig gehalten, da Sie nur zahlen, wenn die Funktion ausgeführt wird. Lambda bietet Ihnen die Flexibilität, jedes Spielfeature als separaten Microservice in einer Programmiersprache Ihrer Wahl zu schreiben. Sie könnten sich beispielsweise dafür entscheiden, .NET-Lambda-Funktionen zu entwickeln, wenn Sie Erfahrung mit C# zur Erstellung von Unity-Spielen haben, oder Sie könnten Lambda-Funktionen für Node.js entwickeln, wenn Sie sowohl ein Frontend als auch ein Backend für ein webbasiertes Spiel programmieren möchten. JavaScript
- **NoSQL Data Store für Spiel- und Spielerdaten:** Verwenden Sie DynamoDB, um Ihre Spieler- und Spieldaten zu speichern, da es speziell für die Speicherung großer Datenmengen aus Microservices entwickelt wurde. Wie in dieser Architektur veranschaulicht, hat es sich bewährt, separate Datenspeicher für die Datenspeicheranforderungen der einzelnen Spielfunktionen zu verwenden, sodass Sie die Funktionen problemlos unabhängig überwachen und verwalten können. Dadurch entstehen auch Trennungsgrenzen, wenn sich innerhalb Ihres Teams die Eigentümerschaft für Funktionen oder Dienste ändert. In dieser Referenzarchitektur werden DynamoDB-Tabellen verwendet, um Daten wie Verbindungsstatus, Spieldetails, Spielerfortschritte und Bestenlisteninformationen zu speichern.
- **Einzelspieler-Gameplay:** Einzelspieler-Funktionen ermöglichen es Spielern, Aktionen wie das Auswählen und Spielen eines Spiels und das Anzeigen der Bestenliste durchzuführen. Diese Funktionen werden als RESTful Backend-Services implementiert, die mit einer Amazon API Gateway-HTTP-API gehostet werden, die die entsprechende Lambda-Funktion aufruft, um Daten in DynamoDB-Tabellen abzurufen und festzulegen. Wenn das Gameplay abgeschlossen ist, sendet das Backend auch Benachrichtigungen an Amazon SNS SNS-Themen, die asynchron Lambda-Funktionen starten, um den Fortschritt und die Statistiken des Spielers zu speichern.
- **Mehrspieler-Gameplay:** Für die Funktionen von Multiplayer-Spielen müssen die Spieler in der Lage sein, mit dem Spiel zu point-to-point kommunizieren sowie Updates von anderen verbundenen Spielern zu senden und zu empfangen. Eine WebSockets Implementierung eignet sich für die point-to-point Kommunikation in einem leichten Spiel, z. B. bei Quizspielen. Spieler

können eine WebSockets Verbindung zu Amazon API Gateway herstellen WebSockets, das die Verbindung verwaltet und die Lambda-Funktionen nur aufruft, wenn Nachrichten für einen Spieler gesendet oder empfangen werden müssen. Für Anwendungsfälle, in denen one-to-many Kommunikation zwischen Spielern erforderlich ist, AWS IoT Core bietet es Unterstützung für Nachrichten WebSockets über MQTT, sodass Kunden Themen abonnieren und auf die empfangenen Nachrichten reagieren können. In dieser Architektur werden WebSockets Over-MQTT verwendet, um Anwendungsfälle wie die Übertragung von Live-Updates im Spiel und das Stellen von Fragen an verbundene Spieler zu unterstützen. Als Alternative können Sie Redis Pub/Sub für die Nachrichtenzustellung oder Redis Streams wählen, wenn Sie die Aufbewahrung von Nachrichten benötigen. AWS IoT

- Verwenden Sie VPC-fähige Lambda-Funktionen, um auf Ressourcen in Ihren privaten Subnetzen zuzugreifen: Konfigurieren Sie VPC-fähige Lambda-Funktionen für den Zugriff auf Ressourcen in den privaten Subnetzen Ihrer VPC, wie Amazon ElastiCache, wodurch die Abfragezeiten für Datensätze mit niedriger Latenz wie Live-Bestenlisten reduziert werden.

[Weitere Informationen finden Sie unter Anleitung für das Backend-Hosting von benutzerdefinierten Spielen unter. AWS](#)

Entwicklung von Cloud-Spielen (CGD)

Cloud-Spieleentwicklung (CGD) bezieht sich auf die Infrastruktur und Tools, die für den Lebenszyklus der Spieleentwicklung erforderlich sind, um ein Spiel zu erstellen, zu testen und zu entwickeln. Die Spieleentwicklung erfolgt in Zusammenarbeit zwischen Benutzern, und die Infrastrukturanforderungen ändern sich häufig in den Entwicklungsphasen.

Viele Spieleentwickler setzen weltweit verteilte und dezentrale Entwicklungsteams ein, weshalb Technologien erforderlich sind, die diese Art der Entwicklung unterstützen. Spieleentwickler können diese Umgebungen ganz oder teilweise dort hosten AWS und deren globale Verfügbarkeit nutzen, um Ressourcen näher an den Benutzern AWS-Regionen zu platzieren und ihre Entwicklungsumgebungen kostengünstiger zu verwalten, indem sie Rechen- und Speicherressourcen nach Bedarf skalieren.

Die Umgebungen können je nach den Bedürfnissen der Spieleentwickler variieren, umfassen jedoch in der Regel Entwickler-Workstations, auf denen Künstler, Designer, Ingenieure, QA-Tester, Auftragnehmer und anderes Personal ihre Arbeit ausführen können. Zu diesen Umgebungen gehört in der Regel auch eine Build-Farm, die aus Quellcode-Repositorys besteht, in die Benutzer ihre

Änderungen einchecken können, und die CI/CD Infrastruktur für das Erstellen, Verpacken und Testen der entwickelten Artefakte.

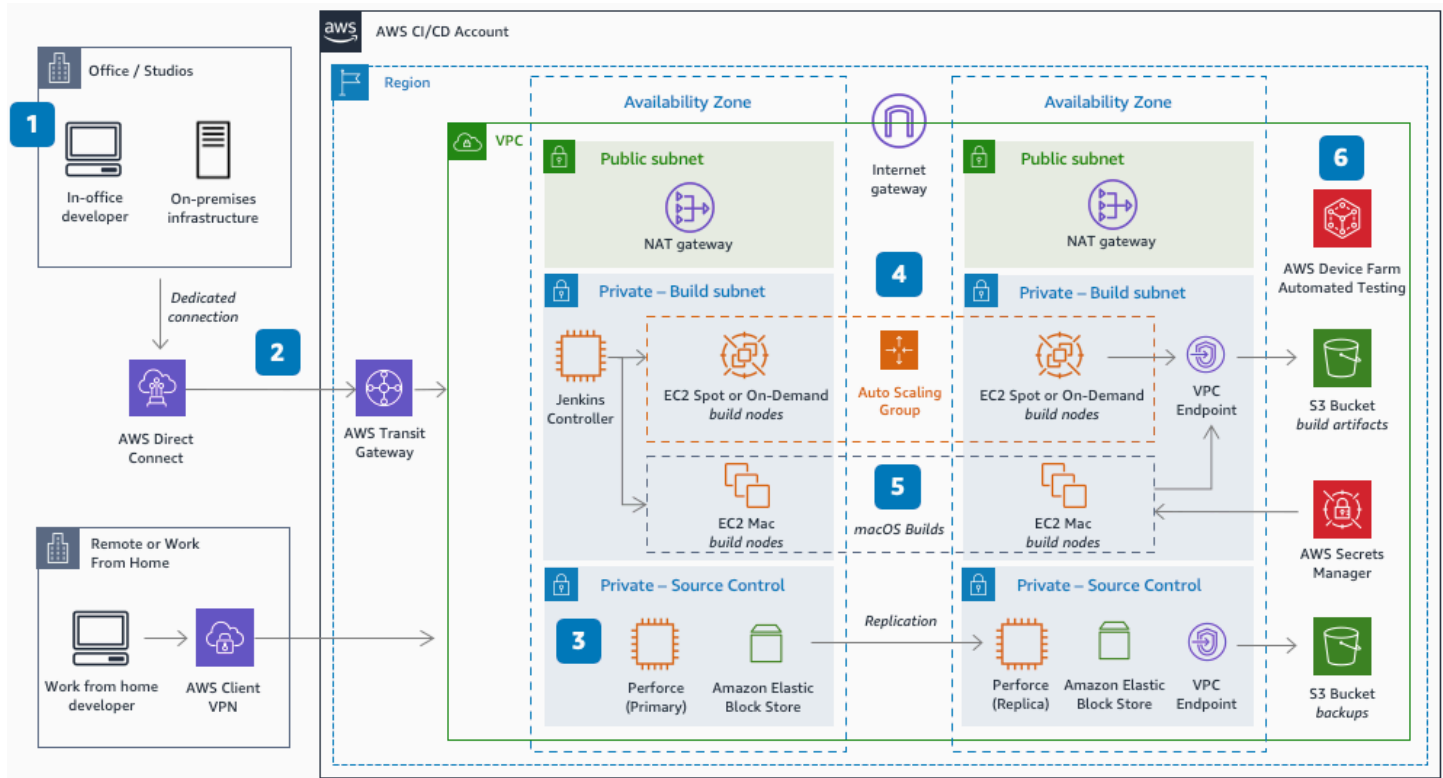
Diese Architekturen für die Spieleproduktion weisen die folgenden Merkmale auf:

- Benutzer sollten über einen Webbrowser oder einen lokalen Desktop-Client wie [Amazon DCV](#) auf eine virtuelle Workstation zugreifen können, der ihnen eine Streaming-Sitzung mit niedriger Latenz bietet, um auf dieselbe Software und Tools zuzugreifen, auf die sie Zugriff hätten, wenn sie an einem Computer in einem Büro oder Entwicklungsstudio arbeiten würden. Diese virtuellen Workstations, in der Regel ein cloudbasierter Server, sollten es Benutzern ermöglichen, zusammenzuarbeiten und an ihren Projekten vollständig in einer Cloud-Umgebung über ein LAN oder WAN zu arbeiten. Wenn Benutzer die Maschinen nicht aktiv nutzen, sollte ihre Arbeit auf einem dauerhaften Cloud-Speicher gesichert werden, z. B. in einem Quellcodeverwaltungs-Repository oder Dateisystem wie [Amazon Elastic File System \(EFS\)](#) und [Amazon FSx](#), und ihr Computer sollte heruntergefahren werden, um die Kosten zu senken.
- [Repositoryys für die Quellcodeverwaltung, wie Perforce, sollten mit hoher Verfügbarkeit konzipiert werden, indem Replikation zwischen Availability Zones oder zwischen lokalen Umgebungen mit Backups, die in Cloud-Speichern wie Amazon S3 gespeichert sind, verwendet werden.](#) Ein cloudbasierter Perforce-Server sollte beispielsweise einen primären Commit-Server enthalten, der in einer Availability Zone gehostet wird und auf einen Standby-Server repliziert wird, der in einer anderen Availability Zone in derselben Region gehostet wird.
- Die Build-Farm-Ressourcen für die Spieleentwicklung sollten mit automatischer Skalierung ausgestattet sein, sodass Rechenressourcen nach Bedarf bereitgestellt werden. [EC2Spot-Instances](#) sollten verwendet werden, um die Kosten zu reduzieren, die bei der Skalierung der Anzahl der für Builds erforderlichen Server anfallen.

Entwicklung von Cloud-Spielen: CI/CD

CI/CD Die Infrastruktur ist bei der Entwicklung von Spielen unabhängig von der Teamgröße wichtig, um die Iterationszeiten zu verbessern, die Zuverlässigkeit der Builds zu verbessern, die Bereitstellung effizienter zu gestalten und den Entwicklungs- und Veröffentlichungsprozess besser zu kontrollieren, um den Spielern ein qualitativ hochwertiges Spielerlebnis zu bieten. Eine CI/CD Spieleentwicklungspipeline besteht in der Regel aus hochverfügbaren Quellcodeverwaltungsservern und Speicher, Rechenressourcen für die Ausführung Ihrer Builds und Software zur Durchführung automatisierter Tests sowie aus der richtigen Netzwerkkonnektivität Ihrer Entwicklungscomputer. Die folgende Referenzarchitektur zeigt, wie Spiele-Builds aus dezentralen oder lokalen

Spielentwicklungsumgebungen in die Cloud ausgelagert werden können, um Entwicklern bei der Migration oder dem Aufbau neuer Build-Farmen AWS Cloud zu helfen.



Laden Sie Spiel-Builds in die Cloud aus

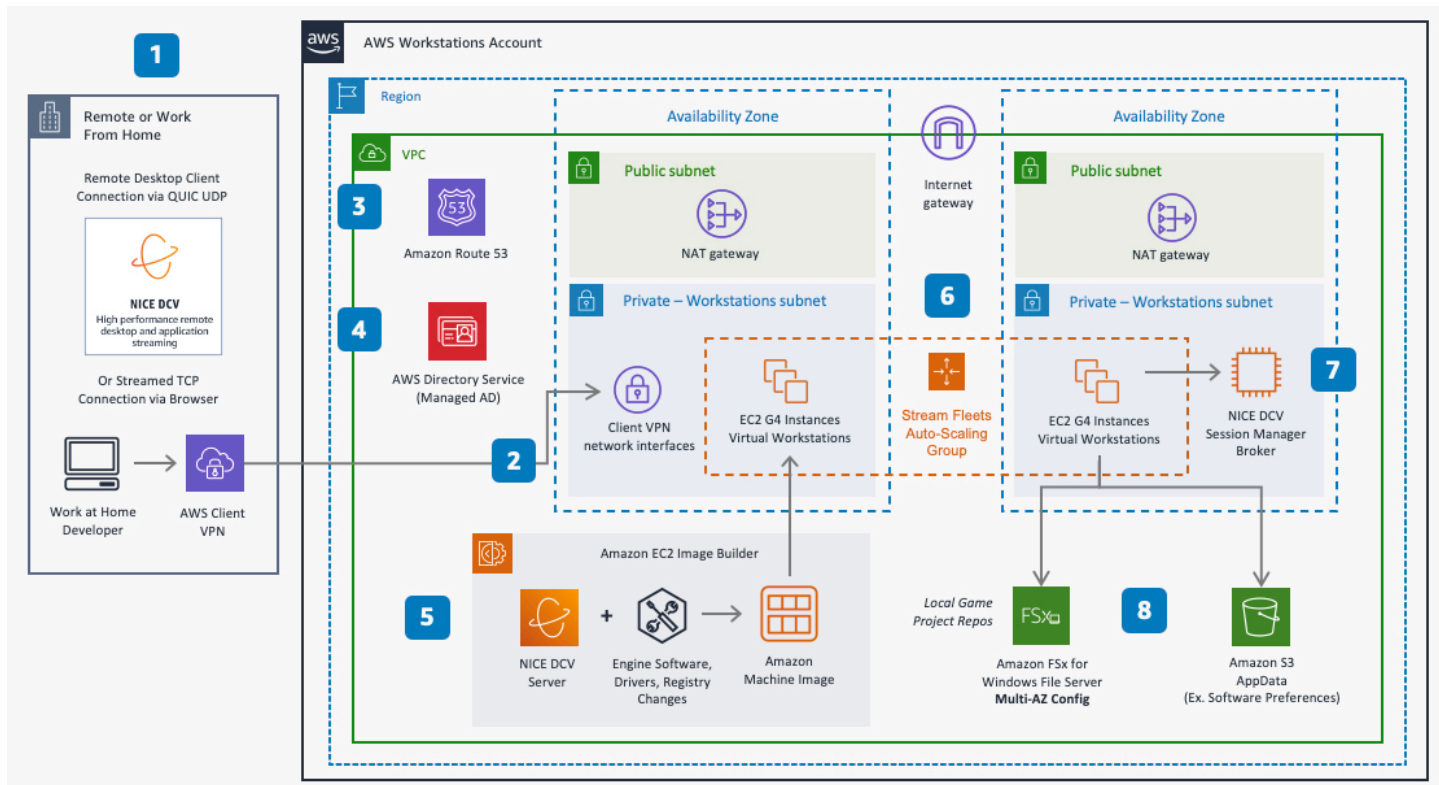
1. AWS Direct Connect bietet eine private, dedizierte Verbindung mit niedriger Latenz AWS für Entwickler im Büro. Remote-Entwickler verwenden Zero-Trust-Technologien wie AWS Verified Access oder virtuelle private Netzwerke (VPN) wie AWS Client VPN.
2. AWS Transit Gateway vereinfacht das Netzwerkmanagement für die Konnektivität zwischen VPCs und von lokalen Netzwerken.
3. Perforce verwaltet die Quell- und Versionskontrolle (CI), die durch Amazon EBS-Speicher unterstützt wird, sodass schnell auf persistente Daten zugegriffen werden kann. Perforce Helix Core ist verfügbar in der AWS Marketplace
4. Commits starten einen Build (CD) in Jenkins, wenn Entwickler Änderungen an Perforce übertragen, die an einen Branch gebunden sind. Perforce initiiert eine POST-Übertragung einer JSON-Payload an Jenkins. Der Jenkins-Controller ruft Headless-CLI-Befehle der Engine auf, um den Build-Prozess auf kurzlebigen Docker-Knoten (wie Amazon Spot-Instances oder Amazon EC2 On-Demand-Instances) auszuführen und zu parallelisieren. EC2 Entwickler können die Verfügbarkeit erhöhen, indem sie zwei Jenkins-Controller, einen in jeder Availability Zone, hinter einem Load Balancer verwenden. Für einige Game-Engines benötigen Entwickler möglicherweise

- eine zusätzliche Lizenzierungsinfrastruktur, die in zusätzlichen Subnetzen konfiguriert ist, um bei jeder Ausführung eines gleichzeitigen Builds Lizenzen für den Build-Kontext zu verkaufen.
5. Der Xcode-Teil der iOS-Builds wird auf Amazon EC2 Mac-Instances ausgelagert, um die IPA-Datei zu signieren, zu erstellen und zu exportieren, wodurch der Prozess aufgeteilt und die Build-Zeiten reduziert werden. AWS Secrets Manager enthält Bereitstellungsprofile, private Schlüssel und Zertifikate.
 6. Build-Artefakte werden an Amazon S3 übermittelt, das Benachrichtigungen über Erfolg oder Misserfolg sendet. AWS Device Farm ermöglicht automatisiertes Testen für mobile Geräte.

Entwicklung von Cloud-Spielen: Workstations

Bei der Spieleentwicklung arbeiten häufig verteilte Teams von verschiedenen Standorten aus. Sie benötigen Zugriff auf eine gemeinsame Infrastruktur und die Möglichkeit, die gemeinsame, geografisch verteilte Entwicklung zu unterstützen. Dieser Trend hin zu einem stärker verteilten Spieleentwicklungsprozess, einschließlich der Nutzung von work-for-hire Studios und Telearbeit, erfordert die Implementierung robuster Technologien und Workflows, um Produktivität, gemeinsames Fachwissen und agile Entwicklungspraktiken zu fördern.

Die folgende Referenzarchitektur zeigt, wie das Amazon AWS DCV-Protokoll zum Hosten von Remote-Workstations für die Spieleentwicklung verwendet wird.



Streamen Sie die Spieleentwicklung mit Amazon DCV von überall

1. Amazon DCV ist ein Streaming-Protokoll, das 4K- und 60-FPS-Streaming unterstützt. Entwickler, die einen Browser verwenden, stellen eine Verbindung über TCP-Verbindungen her, wohingegen Desktop-Clients QUIC UDP über Port 8443 verwenden können, um die Leistung zu erhöhen.
2. Entwickler verwenden AWS Client VPN für eine sichere Verbindung zu Netzwerkschnittstellen in den Workstation-Subnetzen mit Source Network Address Translation (SNAT).
3. Amazon Route 53 bietet privates DNS für die Ressourcen in der VPC sowie eingehende und ausgehende DNS-Weiterleitung.
4. Directory Service bietet verwaltetes Microsoft Active Directory, um lokalen Speicher für Spieleprojekte zu ermöglichen, der einzelnen Benutzern zugeordnet ist.
5. Arbeitsstationen werden mit einem Amazon Machine Image (AMI) erstellt, das mit Image Builder erstellt wurde. Zu den Bildern gehören Amazon DCV Server, Entwicklersoftware, Registrierungsänderungen und Treiber wie NVIDIA-Gaming-Treiber oder Peripherie-Treiber. AWS Marketplace beinhaltet häufig AMIs verwendete Komponenten für Workstations.
6. Workstation-Flotten verwenden Grafiken, die EC2 Amazon-Instance-Typen bereitstellen GPUs, und werden mithilfe von EC2 Auto Scaling-Gruppen skaliert.

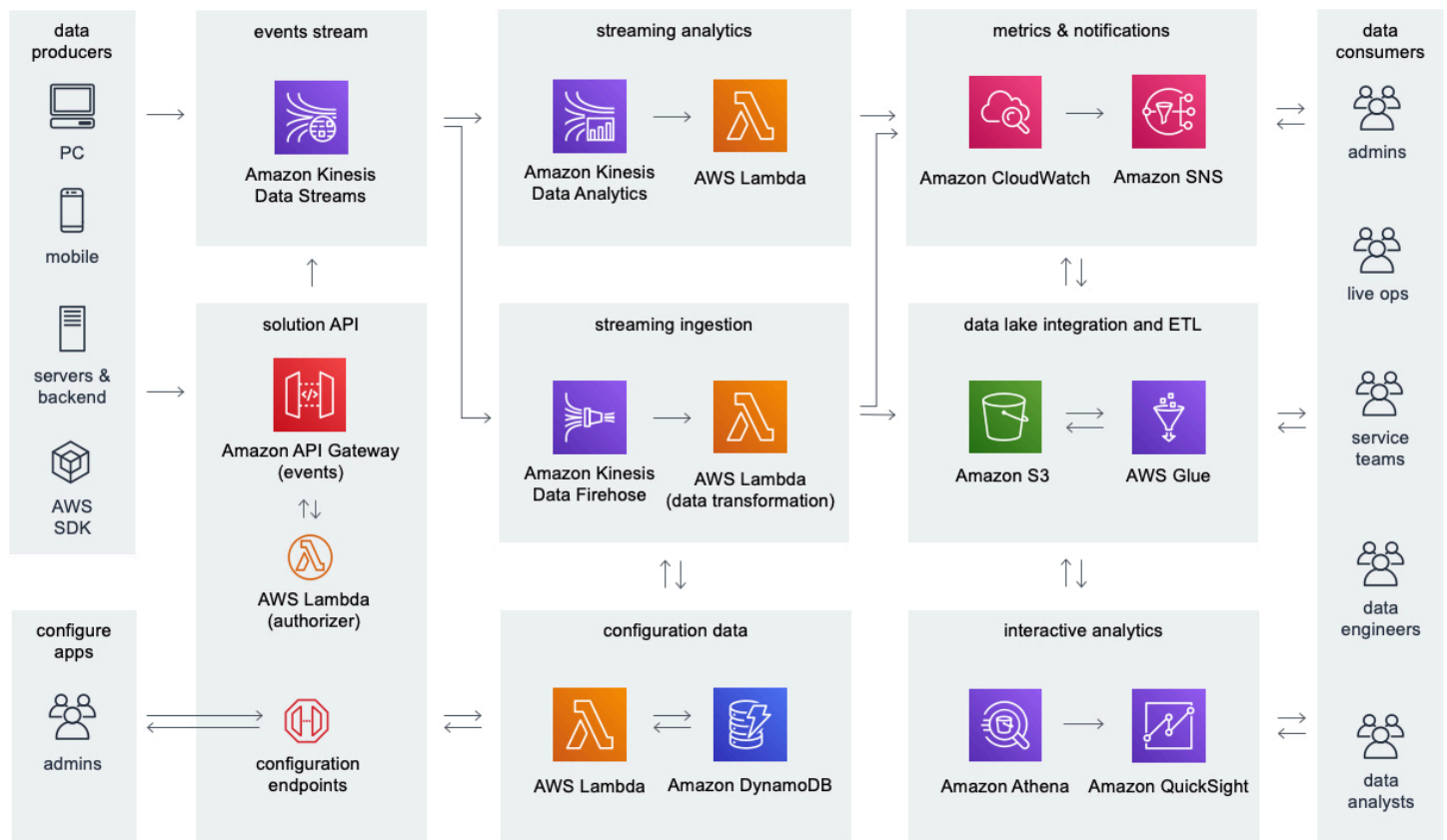
7. Der Amazon DCV Session Manager Broker ermöglicht die Verwaltung von Amazon DCV-Sitzungen.
8. Der lokale Dateispeicher von Projekten wird in Amazon FSx for Windows File Server gehostet. Entwickler verpflichten sich zu einer separaten CI/CD-Pipeline, indem sie vom lokalen Speicher zur Quellcodeverwaltung wechseln.

Spieleanalytik-Pipeline

Spieleentwickler suchen zunehmend nach Möglichkeiten, das Spielerverhalten besser zu verstehen, um das Spielerlebnis zu verbessern und so ihre Spielerbasis zu halten und zu vergrößern. Bei der Spielanalyse handelt es sich um die technische Infrastruktur und die Prozesse, die erforderlich sind, um die Daten zu verstehen und zu analysieren, die aus dem Spiel und den damit verbundenen Diensten generiert werden. Dies erfordert in der Regel die Verwendung einer Analyse-Pipeline-Architektur, die diesen end-to-end Prozess unterstützen kann, wie z. B. die Implementierung der [Game Analytics-Pipeline-Lösung](#).

Architekturen für Spieleanalysen weisen die folgenden Merkmale auf:

- Datenquellen senden Daten in einem gängigen Format wie JSON und umfassen in der Regel Spieleserver und Spiele-Backend-Dienste sowie Spieleclients, einschließlich PCs, Mobilgeräten und Spielekonsolen.
- Eine Spielanalyse-Pipeline automatisiert den gesamten Arbeitsablauf von der Aufnahme und Speicherung der Rohdaten bis hin zur Verarbeitung in nutzbare Ausgabeformate, sodass sie von Datenverbrauchern wie Endbenutzern und Analyseanwendungen effizient und kostengünstig analysiert werden können.
- Pipelines zur Spielanalyse bieten Unterstützung für die Aufnahme und Verarbeitung großer Mengen an Echtzeitdaten, um sie an das Wachstum eines Spiels anzupassen.
- Bieten Unterstützung für Anwendungsfälle sowohl für Echtzeit- als auch für Batch-Berichte. Beispielsweise werden Echtzeit-Dashboards und Warnmeldungen in der Regel von Live-Ops-Teams verwendet, um die Spielinfrastruktur und das Spielerverhalten zu überwachen und Probleme zu erkennen. Datenanalyseteams verlassen sich in der Regel bei Bedarf auf Batch-Berichte, um Trends im Laufe der Zeit zu verstehen.



Serverlose Spieleanalyse-Pipeline für Gameplay-Telemetrie

Spieldaten werden von Spieleclients, Spieleservern und anderen Anwendungen aufgenommen. Die Streaming-Daten werden für die Data-Lake-Integration und interaktive Analysen in Amazon S3 aufgenommen. Streaming-Analysen verarbeiten Ereignisse in Echtzeit und generieren Metriken. Datenkonsumenten analysieren Metrikdaten in Amazon CloudWatch und Rohereignisse in Amazon S3.

- **Lösungs-API und Konfigurationsdaten:** Verwenden Sie Amazon API Gateway, um eine REST-API zur Verwaltung Ihrer Spielanalyse-Pipeline und zum Speichern von Konfigurationsdaten in Amazon DynamoDB mithilfe von Lambda-Funktionen bereitzustellen. Sie können auf dieser API ein internes Portal oder eine benutzerdefinierte Befehlszeilschnittstelle für die Verwaltung einrichten. Eine REST-API bietet auch Serverauthentifizierung für die Aufnahme von Gameplay-Daten aus Datenquellen und die Weiterleitung der Telemetriedaten an Amazon Kinesis Data Streams zur Echtzeitverarbeitung und Aufnahme in den Speicher.
- **Event-Stream:** Amazon Kinesis Data Streams erfasst Streaming-Daten aus Ihrem Spiel und ermöglicht die Datenverarbeitung in Echtzeit durch Amazon Data Firehose und Amazon Managed Service für Apache Flink.

- **Streaming-Analyse:** Managed Service für Apache Flink analysiert Streaming-Ereignisdaten aus den Kinesis Data Streams und kann benutzerdefinierte Metriken und Warnungen generieren, die CloudWatch mithilfe von Lambda-Funktionen veröffentlicht werden.
- **Metriken und Benachrichtigungen:** Verwenden Sie Amazon CloudWatch , um die Metriken, Protokolle und Alarme Ihrer Lösung zu überwachen. Verwenden Sie Amazon SNS, um Benachrichtigungen an Bereitschaftsmitarbeiter und andere Datenverbraucher zu senden.
- **Streaming-Aufnahme:** Verwenden Sie Firehose, um Ihre Streaming-Daten aus Kinesis Data Streams zu nutzen und sie zur langfristigen Speicherung, Transformation und Integration mit anderen Daten an Ihren Data Lake in Amazon S3 zu übertragen.
- **Data-Lake-Integration und ETL:** Verwenden Sie es AWS Glue für ETL-Verarbeitungsworkflows und zur Organisation Ihrer Metadaten in der AWS AWS Glue Data Catalog. Dies bildet die Grundlage für einen Data Lake für die Integration mit flexiblen Analysetools.
- **Interaktive Analysen:** Endbenutzer können Amazon Athena verwenden, um interaktive Ad-hoc-Abfragen zu den in Amazon S3 gespeicherten Datensätzen durchzuführen, und Quick Suite kann zur Erstellung von Dashboards verwendet werden.

In der [Game Analytics-Pipeline](#) finden Sie eine automatisierte Referenzimplementierung einer Analyse-Pipeline, die mithilfe von in Ihrem Konto bereitgestellt werden kann. AWS CloudFormation

Definitionen

Das AWS Well-Architected Framework basiert auf sechs Säulen: betriebliche Exzellenz, Sicherheit, Zuverlässigkeit, Leistungseffizienz, Kostenoptimierung und Nachhaltigkeit. AWS bietet mehrere Kernkomponenten, mit denen Sie state-of-the-art Architekturen für Ihre Spiele-Workloads entwerfen können. In diesem Abschnitt geben wir einen Überblick über die wichtigsten Definitionen.

Für die Zwecke dieses paper umfasst eine Spielarchitektur die technische Backend-Infrastruktur, die für die Entwicklung und den Betrieb eines Spiels erforderlich ist. Einige Spiele verfügen möglicherweise nicht über soziale Funktionen, Multiplayer- oder andere Online-Funktionen und erfordern möglicherweise nicht die Verwendung bestimmter Aspekte der technischen Backend-Infrastruktur, die in diesem paper beschrieben werden. Eine ausführliche Erläuterung der verschiedenen Arten von Workloads, die häufig zur Unterstützung einer Spielarchitektur eingesetzt werden, finden Sie unter Szenarien.

Die AWS Cloud Infrastruktur basiert auf Regionen und Availability Zones.

- Eine Region ist ein physischer Standort auf der Welt, an dem wir mehrere Availability Zones haben.
- Availability Zones bestehen aus einem oder mehreren diskreten Rechenzentren, die jeweils über redundante Stromversorgung, Netzwerke und Konnektivität verfügen und in separaten Einrichtungen untergebracht sind.

Abhängig von den Eigenschaften Ihres Spiels möchten Sie möglicherweise bestimmte Komponenten Ihrer Spielarchitektur in mehreren Regionen einsetzen, um beispielsweise die Leistung für Spieler zu verbessern oder Spielern je nach Standort maßgeschneiderte Erlebnisse zu bieten.

Es gibt viele verschiedene Arten von Spielen, und die technische Backend-Infrastruktur, die zur Unterstützung eines Spiels erforderlich ist, hängt von der Art des Spiels ab, das entwickelt wird. Zu den beliebten Arten von Spielen können beispielsweise Ego-Shooter (FPS), Rollenspiele (RPG), MMOG (Massively Multiplayer Online Games), Battle Royales (BR), Sportspiele, Puzzlespiele und mehr gehören. Es gibt auch verschiedene Spielinteraktionsmodi, die die Architektur des Spiels beeinflussen, z. B. rundenbasiertes und gleichzeitiges Spielen mit unterschiedlichen Leistungsmerkmalen.

Spiele wurden so entwickelt, dass sie auf einem oder mehreren Spielesystemen gespielt werden können, darunter Desktop-, Web-, Mobilgeräte, Konsolen und neueren Interaktionsmodi wie Augmented Reality (AR), Virtual Reality (VR) und Game-Streaming-Lösungen. Spiele unterstützen

in der Regel systemübergreifendes Gameplay, was bedeutet, dass Spieler ihren Spielfortschritt speichern und das Spiel auf anderen Systemen fortsetzen sowie Spielsitzungen mit Spielern auf anderen Systemen starten können.

Die Monetarisierung von Videospiele ermöglicht es Spieleverlagen, Einnahmen zu erzielen, indem sie verschiedene Strategien wie Werbung, digitale und Einzelhandelskäufe, In-Game-Käufe von herunterladbaren Inhalten (DLC), sogenannten Mikrotransaktionen, und durch kostenpflichtige Abonnements, die für das Spielen des Spiels erforderlich sind, einsetzen. Zu den gängigsten Leistungsindikatoren (KPIs) in der Spielebranche gehören:

- Aktive Benutzer pro Tag (DAU)
- Aktive Benutzer pro Monat (MAU)
- Gleichzeitige Benutzer (CCU)
- Sitzungsdauer
- Kosten pro Installation (CPI)
- Lebenszeitwert des Spielers (LTV)
- Durchschnittlicher Umsatz pro Nutzer (ARPU)

Spielsystem

Videospiele wurden für das Spielen auf einem Spielesystem entwickelt, das Bedienelemente für die Client-Eingabe, Grafiken, Client-Software (bekannt als Spieleclient) und Hardware sowie in einigen Fällen systemexklusive Funktionen zur Unterstützung des Gameplays bietet.

Spielesysteme werden im Allgemeinen in die folgenden Kategorien unterteilt:

- Konsolen: Speziell entwickelte Unterhaltungssysteme, die für das Spielen von Spielen entwickelt wurden, darunter beliebte Beispiele wie Sony PlayStation, Microsoft Xbox und Nintendo Switch. Konsolen bieten die Möglichkeit, Spiele zu spielen, indem physische oder digital verteilte Spielinhalte auf der vom Spielesystemanbieter hergestellten Konsolenhardware installiert werden. In dieser Definition kann eine Konsole tragbar oder stationär sein und für den Einsatz in einem Home-Entertainment-Szenario vorgesehen sein.
- Personal Computer (PC): Spiele, die mit Computersoftware gespielt werden, die auf einem Client-Computer installiert ist und vom Spieler angepasst werden kann. Aus diesem Grund sind PC-Spiele bei Spielern aufgrund der Flexibilität und Kontrolle, die sie bieten, beliebt.

- **Web:** Spiele, die so konzipiert sind, dass sie mit einem Webbrowser gespielt werden können und die in der Regel den Vorteil bieten, dass ein Spieler unabhängig von seinem Betriebssystem auf das Spiel zugreifen kann.
- **Mobilgeräte:** Spiele, die für das Spielen auf einem Mobiltelefon, in der Regel einem Smartphone-Betriebssystem, entwickelt wurden. Handyspiele werden normalerweise aus einem digitalen App Store heruntergeladen und auf dem Telefon installiert.

Zusätzlich zu den zuvor genannten Systemen gibt es auch Systeme im Entstehen, die noch relativ neu sind und weiter wachsen und im Vergleich zu den vorherrschenden Systemen einen viel geringeren Marktanteil haben. Beispiele für Spielesysteme in dieser Kategorie sind AR, VR und Game-Streaming, manchmal auch als Cloud-Gaming bezeichnet.

Beim Game-Streaming wird das Gameplay in der Cloud gerendert und auf einen Thin Client, in der Regel einen Browser, gestreamt. Game-Streaming ermöglicht es einem Spieler, ein Spiel zu spielen, das vollständig remote gehostet wird, typischerweise in der Cloud von einem Game-Streaming-Dienstanbieter. Beim Game-Streaming stellt der Spieler über einen Browser oder einen Thin Client, der vom Cloud-Gaming-Dienstanbieter (Spielesystem) bereitgestellt wird, eine Verbindung zu einem cloudbasierten Spiel her.

Spieleserver

Spieleserver stellen einen der wichtigsten Aspekte der Recheninfrastruktur für Ihr Spiel dar. Spieleserver, manchmal auch als dedizierte Spieleserver bezeichnet, werden verwendet, wenn ein Multiplayer-Spiel entwickelt wird oder wenn eine servergestützte Verarbeitung von Gameplay-Ereignissen erforderlich ist. Der Spieleserver steht im Mittelpunkt der Spielarchitektur und dient als Standort, an dem die Kernlogik ausgeführt wird. Dazu gehören die Verwaltung des Spieler- und Spielstatus sowie die Verwaltung der Interaktionen zwischen den verbundenen Spielclients und dem Spieleserver. Der Spieleserver ist normalerweise einer der leistungsempfindlichsten Aspekte einer Spielarchitektur, da er dafür verantwortlich ist, die Eingaben vom Spielclient eines Spielers zu verarbeiten und sie ordnungsgemäß in Echtzeit an andere verbundene Spieler zu verteilen. Ein Spieleserver mit schlechter Leistung beeinträchtigt die Gesamtleistung des Spielerlebnisses. Daher solltest du die Leistung des Spieleservers optimieren und ausreichend Kapazität bereitstellen, insbesondere zu Spielstarts oder zu Spitzenzeiten des Spiels.

Für die Zwecke dieses Dokuments bezieht sich ein Spieleserver oder eine Spieleserver-Instanz auf den Computer, z. B. eine virtuelle Maschine, der einen oder mehrere Spieleserverprozesse hostet. Ein Gameserver-Prozess stellt eine einzelne Instanz Ihres Gameserver-Builds dar, die eine Spielsitzung

hostet. Dabei handelt es sich um eine Instanz Ihres laufenden Spiels, mit der sich Spieler über eine Spielersitzung verbinden können. Aus diesem Grund beziehen wir uns oft synonym auf Gameserver-Prozess oder Spielsitzung, da es eine direkte Beziehung zwischen einer Spielsitzung und dem Gameserver-Prozess gibt, der sie hostet. In gibt es mehrere Optionen für die Datenverarbeitung AWS, um Spieleserver zu hosten, die durch die elastische Bereitstellung von Ressourcen Zugriff auf skalierbare Cloud-basierte Kapazitäten bieten.

Amazon EC2 bietet cloudbasierte virtuelle Server, sogenannte Instances, mit Unterstützung für mehrere Versionen von Linux und Windows. Sie können Instances erstellen und sie wie einen anderen Server oder eine virtuelle Maschine direkt verwalten. In der Regel werden mehrere Spieleserverprozesse auf einer Instanz bereitgestellt, um die Effizienz zu verbessern und die Kosten zu senken. Amazon EC2 ist eine gute Wahl für Spieleserver, wenn Sie die größtmögliche Kontrolle über die Computerinfrastruktur wünschen.

Amazon GameLift bietet eine vollständig verwaltete Lösung für das Hosting von dedizierten Spieleservern in der Cloud sowie zusätzliche Funktionen wie Matchmaking mit GameLift FlexMatch. GameLift bietet zusätzlich EC2 zu Amazon eine Abstraktionsebene, um die Verwaltung von Spieleservern zu vereinfachen, und ist in den meisten Fällen verfügbar, AWS-Regionen sodass Sie Spieleserver in der Nähe von Spielern hosten können, um die Latenz zu reduzieren, eine hohe Verfügbarkeit zu erreichen und die Kosten durch die Verwendung von Spot-Instances erheblich zu senken. Es GameLift kann zwar in bestehende Spiele-Backends integriert werden, ist aber besonders nützlich für Spieleentwickler, die keine eigenen Gameserver-Management- und Matchmaking-Lösungen entwickeln möchten und eine Lösung bevorzugen, die von ihrem Spiel verwaltet wird AWS und skaliert werden kann, wenn ihr Spiel wächst.

Amazon Elastic Container Service (Amazon ECS) ist ein vollständig verwalteter Container-Orchestrierungsservice, mit dem Sie Docker-basierte Container ausführen können. Sie können auch Amazon Elastic Kubernetes Service (Amazon EKS) verwenden, um Docker-basierte Container auszuführen, die mit Kubernetes erstellt wurden. Die Verwendung von Container-Technologien, wie sie von Amazon ECS und Amazon EKS bereitgestellt werden, kann Ihnen helfen, Ihre Rechenauslastung zu verbessern, indem Sie viele Spieleserverprozesse oder andere Spielanwendungsinstanzen effizient in eine EC2 Instance packen.

Die Verwendung von Containern kann auch die Produktivität von Entwicklern verbessern, indem Anwendungen mit derselben Docker-Image-Betriebslaufzeit gehostet werden, die Entwickler während der Entwicklung auf ihren lokalen Computern verwendet haben. Sie können den Betriebsaufwand weiter reduzieren AWS Fargate, indem Sie, eine serverlose Rechenlösung für den Betrieb von Containern, verwenden, die sowohl mit Amazon EKS als auch mit Amazon ECS kompatibel ist.

Fargate eignet sich am besten für Anwendungsfälle, in denen Sie Spieleserver in Containern ausführen möchten, ohne dafür verantwortlich zu sein, die zugrunde liegenden Instanzen zu betreiben, auf denen die Container laufen.

Sie können AWS Outposts damit AWS Infrastruktur und Dienste in einem Rechenzentrum oder einer lokalen Einrichtung ausführen, sodass Spiele in lokalen Umgebungen ausgeführt werden können und dieselbe Infrastruktur zur Unterstützung einer Hybrid-Cloud-Einführungsstrategie AWS verwendet werden kann. AWS Local Zones dienen als Erweiterungen AWS-Regionen, die es Ihren Spieleservern und anderen latenzempfindlichen Workloads ermöglichen, näher an Ihren Spielern oder Entwicklungsteams zu arbeiten. Um die globale Netzwerklatenz für Ihre Spieleserver zu reduzieren, können Sie außerdem AWS Global Accelerator verwenden, um die Leistung des Spielerverkehrs zu Ihren Spieleservern zu verbessern.

AWS Lambda ist ein serverloser Rechendienst, der Code ausführt, ohne Server bereitzustellen oder zu verwalten, was ihn für asynchrone Spieleserver-Anwendungsfälle wie rundenbasierte Spiele oder solche mit geringen Rechenanforderungen und einer kleinen Codebasis nützlich macht und bei denen Gameplay-Funktionen mithilfe einer statusfreien Microservices-Architektur entworfen werden können. Es ist wichtig zu bedenken, dass Lambda-Funktionen ereignisgesteuert und pro Anfrage ausgeführt werden, anstatt einen lang andauernden Spieleserverprozess auszuführen. Lambda bietet die meisten Laufzeitabstraktionen der Optionen in diesem paper, da die zugrunde liegende Anwendung für Entwickler leicht verfügbar ist, aus der sie wählen können, um ihren Code zu hosten.

Bei der Auswahl Ihres Ansatzes für das Hosting von Spieleservern sollten Sie verschiedene Anforderungen berücksichtigen, darunter den betrieblichen Aufwand, ältere Codebasen, Leistungsanforderungen und Skalierbarkeit. EC2Instances und Container sind gute Optionen für ältere Codebasen, da sie die geringste Änderung erfordern, um in die Cloud zu wechseln. Sie können EC2 Instances verwenden, um die Ressourcen einer Recheninstanz zuzuweisen, während Container die Verwaltung und eine hohe Auslastung vereinfachen können. Serverlose Funktionen bieten die höchste Abstraktionsebene, mit der Sie Code definieren können, der nur als Reaktion auf Ereignisse ausgeführt wird, wodurch die Kosten gesenkt werden können.

Spiele-Client

Der Spielclient stellt das Software- und Hardwaregerät dar, das der Spieler zum Spielen eines Spiels verwendet. Der Spielclient stellt die Software bereit, mit der die Eingaben des Spielers in Nachrichten übersetzt werden, die zur Verarbeitung an einen Server gesendet werden, und er ist dafür verantwortlich, die eingehenden Antworten vom Server zu verarbeiten und Ausgaben wie Grafiken für den Spieler wiederzugeben. Bei vernetzten Mehrspieler-Spielen in Echtzeit unterhält der

Spielclient normalerweise für die Dauer einer Spielsitzung eine permanente Netzwerkverbindung zu einem Spieleserver, um die Netzwerklatenz zu reduzieren und die Verarbeitungszeit zu minimieren. Der Spielclient kann jedoch auch über REST mit einem Spieleserver oder Backend-Diensten interagieren.

Messaging

In Spielen gibt es in der Regel drei Hauptkategorien von Nachrichten:

- Nachrichten zur Spielerbindung, die sich an einen bestimmten Nutzer oder eine bestimmte Benutzerkohorte richten, z. B. Spieleinladungen oder Push-Benachrichtigungen
- Gruppennachrichten zwischen Spielern, z. B. Chat im Spiel
- service-to-service-Messaging, z. B. JSON-Nachrichten, die zur Integration von zwei oder mehr Anwendungen verwendet werden

Eine gängige Strategie für das Senden und Empfangen dieser Arten von Nachrichten besteht in der Verwendung von Architekturmustern für Publisher-Subscriber und asynchrone Verarbeitung. AWS bietet mehrere Dienste, die Sie bei der Implementierung von Messaging in Ihrem Spiel unterstützen können.

- Amazon Simple Notification Service (SNS): Verwalteter Service für die Zustellung von Nachrichten zwischen Herausgebern und Abonnenten mithilfe eines pub/sub Architekturmusters. Herausgeber senden Nachrichten über eine API an Amazon SNS, das die Nachrichten asynchron an abonnierte Anwendungen übermittelt und Push-Benachrichtigungen direkt an mobile Clients oder Desktops senden kann, wobei einige der am häufigsten verwendeten Push-Benachrichtigungsdienste unterstützt werden. Amazon SNS kann sowohl für Push-Benachrichtigungen an Kunden als auch für service-to-service Messaging-Anwendungsfälle verwendet werden.
- Amazon Simple Queue Service (SQS): Ein vollständig verwalteter Warteschlangenservice, der es einfach macht, Spieleserver und Ihr Spiel zu integrieren, unabhängig von der jeweils verwendeten Programmiersprache. Viele Aufgaben im Spiel können entkoppelt und im Hintergrund erledigt werden, z. B. das Aktualisieren einer Bestenliste oder der Spielzeitwerte in einer Datenbank. Dieser Ansatz ist eine effektive Methode, um verschiedene Teile Ihres Spiels zu entkoppeln und die Funktionen für Spieler unabhängig von der Backend-Verarbeitung zu skalieren.
- Amazon Managed Streaming for Apache Kafka (MSK): Ein vollständig verwalteter Service, der die Erstellung von Datenstreaming- und Produzenten- oder Verbraucheranwendungen mithilfe von Apache Kafka, einer beliebten Open-Source-Lösung, vereinfacht. Kafka wird in der Regel

für die Aufnahme und Verarbeitung von Echtzeit-Streaming-Daten verwendet und kann für Nachrichtenübermittlungen verwendet werden. service-to-service

- Amazon ElastiCache (Redis OSS): Bietet einen vollständig verwalteten In-Memory-Datenspeicher, der Unterstützung für die beliebte pub/sub Funktion von Redis bietet, die häufig für die Entwicklung von Chatroom-Anwendungen und Hochleistungs-Messaging verwendet wird. service-to-service
Redis unterstützt auch umfangreiche Datentypen wie Listen und Gruppen, sodass Entwickler Redis für Hochleistungswarteschlangen verwenden können.
- Amazon Pinpoint: Bietet Nachrichten zur Benutzerinteraktion per E-Mail, SMS, Sprach- und Push-Benachrichtigungen. Amazon Pinpoint kann beispielsweise verwendet werden, um Spielern Nachrichten zur Benutzerinteraktion zu senden, um sie wieder zum Spiel einzuladen, und kann für transaktionale Anwendungsfälle wie die Unterstützung von Multi-Faktor-Authentifizierungstoken, Auftragsbestätigungen und E-Mails zum Zurücksetzen von Passwörtern verwendet werden.

Live-Spielbetrieb (Live Ops)

Live-Spielbetrieb (Live Ops) ist eine Art der Spielverwaltung und des Betriebs, bei der ein Spiel als Live-Service behandelt wird und kontinuierlich neue Funktionen, Updates, Werbeaktionen, In-Game-Events und Verbesserungen für das veröffentlichte Spiel bereitgestellt werden, um das Spielerlebnis für die Spielergemeinschaft zu verbessern.

Traditionell wurden Spiele als Produkte und nicht als Dienstleistungen bereitgestellt, und neue Inhalte und Funktionen wurden häufig eher in nachfolgende Veröffentlichungen oder Fortsetzungen als in das eingeführte Produkt integriert. Mit einem Live-Ops-Ansatz für das Spielmanagement kann ein Game Operations Team ein Spiel auf den Markt bringen und durch Experimente, Werbeaktionen, In-Game-Events und Innovationen eine engagierte Spielergemeinschaft aufrechterhalten, um die Spieler zu unterhalten.

Dieser Ansatz hat zwar den Vorteil, neue Strategien zur Spielerbindung zu erschließen und wiederkehrende Einnahmequellen zu erschließen, erfordert jedoch mehr operatives Fachwissen. Um beispielsweise eine erfolgreiche Live-Ops-Strategie umzusetzen, muss ein Entwickler möglicherweise Cloud-Dienste integrieren oder seine eigene technische Backend-Infrastruktur betreiben. Sie benötigen auch eine effektive Methode, um Probleme zu identifizieren und darauf zu reagieren, die im Spiel oder in der Spielergemeinschaft auftreten und sich negativ auf das Spielerlebnis auswirken können.

Operative Exzellenz

Der Schwerpunkt Operational Excellence konzentriert sich auf bewährte Verfahren für die Bereitstellung und den Betrieb von cloudbasierten Spielen in großem Maßstab. Es ist wichtig, sich auf operative Exzellenz zu konzentrieren, um ein positives Spielerlebnis zu gewährleisten und präventive Maßnahmen zu ergreifen, um sich auf Probleme vorzubereiten und diese zu beheben, die sich auf ihr Spielerlebnis auswirken.

Schwerpunktbereiche

- [Designprinzipien](#)
- [Live-Betrieb](#)
- [Struktur des Kontos](#)
- [Bereitstellungen von Spielen](#)
- [Überwachung Health Gesundheitszustands](#)
- [Lasttest](#)
- [Optimierung im Laufe der Zeit](#)
- [Ressourcen](#)

Designprinzipien

Zusätzlich zu den Designprinzipien aus dem Whitepaper Well-Architected Framework können Ihnen die folgenden Designprinzipien dabei helfen, beim Entwickeln und Betreiben von Spielen operative Exzellenz zu erreichen:

- Definieren Sie messbare und erreichbare Ziele für die Spielbetriebsteams und passen Sie sie gegebenenfalls an: Aufgrund der Trefferabhängigkeit von Spielen ist es schwierig, im Voraus zu bestimmen, wie viele Spieler Ihr Spiel spielen werden, wenn es veröffentlicht wird, oder welche Erwartungen die Spieler an Ihren laufenden Spielbetrieb haben werden. Es ist wichtig, gemeinsam mit den Stakeholdern ehrgeizige, aber erreichbare Ziele zu setzen und einen Ansatz zu entwickeln, der skaliert werden kann, wenn Ihr Spiel die Prognosen übertrifft, und herunterskaliert werden kann, während die Spieleentwicklungsteams das Spielerlebnis optimieren. Bereite dich angemessen vor und teste sie im Voraus, um diese Anforderungen zu erfüllen und deine geschäftlichen und technischen Stakeholder auf die Zielvorgaben für den Betrieb des Spiels abzustimmen. Wenn die Ziele definiert sind, können die Spielteams bei der Planung, dem Design,

der Bereitstellung, dem Testen, der Bereitstellung und dem Betrieb der Backend-Infrastruktur des Spiels ein angemessenes Gleichgewicht zwischen Kosten und Leistung erreichen.

- Verwenden Sie operative Runbooks, um Skalierungsaktivitäten im Zusammenhang mit Spieleinführungen und besonderen Ereignissen zu planen: Die Teams für den Spielbetrieb sollten sich mit den Geschäftsbeteiligten abstimmen, um Prognosen für die zu erwartende maximale Parallelität der Spieler bei Veranstaltungen zu modellieren und eine proaktive Planung durchzuführen, um die Infrastrukturkapazität im Voraus zu skalieren. Aufgrund des schwankenden Spielerverkehrs während Events sollten Sie Ihre bestehenden automatisierten Skalierungssysteme im Voraus planen und vorab skalieren, um Ihre Erfolgchancen bei einem Event zu erhöhen und sicherzustellen, dass Sie über genügend Ressourcen verfügen, um den Spielern ein positives Spielerlebnis zu bieten. Implementieren Sie Performance-Engineering-Praktiken, um sich einen Überblick über Ihre Ressourcen und ein datengestütztes Verständnis der Kapazität Ihres Systems zu verschaffen, das Ihnen als Leitfaden für Aktivitäten vor der Skalierung und automatische Skalierungskonfigurationen dienen kann. Entwickeln Sie operative Runbooks, um für Konsistenz im Prozess zu sorgen. Diese frühzeitige Planung und die Reaktion auf die Nachfrage der Spieler sind besonders wichtig für Live-Spiele, bei denen eine zuverlässige Leistung und Infrastruktur gewährleistet sein muss, um eine aktive, engagierte Spielerbasis über einen längeren Zeitraum zu unterstützen.
- Etablieren Sie ein Betriebsmodell für die Entgegennahme, Untersuchung und Beantwortung von Support-Anfragen von Spielern: Überwachen Sie nach dem Start die Berichte über Beschwerden und Probleme mit dem Spiel. Implementieren Sie geeignete Systeme, um auf sichere und effektive Weise mit Spielern zu interagieren und Spielerprobleme angemessen zu lösen, wie z. B. Community-Foren, soziale Medien, E-Mail, Ticketsysteme, Call Center oder automatisierte Chat-Bot-Lösungen. Dies ist besonders wichtig für Live-Service-Spiele, die eine kontinuierliche Kommunikation mit der Spielerbasis, die Reaktion auf das Feedback der Spieler, um den sich ändernden Bedürfnissen gerecht zu werden, und die Aufrechterhaltung einer engagierten Community über einen längeren Zeitraum erfordern.

Live-Betrieb

GAMEOPS01: Wie definierst du die Live-Operations-Strategie (Live Ops) deines Spiels?

Entwickeln Sie in Absprache mit den Geschäftsbeteiligten eine Strategie für den Live-Betrieb (Live Ops) für Ihr Spiel auf der Grundlage definierter Ziele und Leistungskennzahlen.

Best Practices

- [GAMEOPS01-BP01 Verwenden Sie Spielziele und Kennzahlen zur Unternehmensleistung, um Ihre Live-Betriebsstrategie zu entwickeln](#)

GAMEOPS01-BP01 Verwenden Sie Spielziele und Kennzahlen zur Unternehmensleistung, um Ihre Live-Betriebsstrategie zu entwickeln

Konsultieren Sie Interessenvertreter aus der Wirtschaft, wie Spielehersteller und Verlagspartner, um Ziele und Leistungskennzahlen für ein Spiel festzulegen. Dies kann dir helfen, Pläne für die Verwaltung des Spiels zu entwickeln, einschließlich der Festlegung deiner Wartungsfenster, der Zeitpläne für Software- und Infrastruktur-Updates sowie der Ziele für Systemzuverlässigkeit und Wiederherstellbarkeit.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Anhand dieser Kennzahlen können Sie auch bestimmen, in welcher Phase des Lebenszyklus Ihres Spiels Sie einen Live-Operationsteam (Live Ops) einrichten sollten, um den Zustand des Spiels zu überwachen, direktes Feedback zum Spiel zu sammeln und optimierte und automatisierte Veröffentlichungsprozesse zu entwickeln. Ein neues Spiel könnte zum Beispiel warten, bis ein bestimmter Umfang erreicht ist, gemessen an der Anzahl der aktiven Spieler, dem Umsatz oder anderen Kennzahlen, bevor ein eigenes Live-Operations-Team eingerichtet wird. Ein etabliertes Spieleentwicklungsstudio verfügt möglicherweise bereits über Erfahrung im Live-Betrieb, vielleicht für seine anderen Spiele, sodass es nur das neue Spiel einbauen müsste.

Implementierungsschritte

- Sie können die Ziele für Spielergleichheit (CCU) und tägliche und monatliche aktive Nutzer (DAU und MAU) festlegen, die die Spieleinfrastruktur effektiv unterstützen kann, Ihre Infrastrukturbudgets, Ihre finanziellen Ziele und andere Leistungsziele definieren, z. B. die Häufigkeit der Veröffentlichung von Inhalten und Funktionen zur Steigerung der Spielerbindung. Diese Ziele und Kennzahlen fließen in Entscheidungen über das Spieldesign, das Veröffentlichungsmanagement, die Beobachtbarkeit und die Unterstützung ein, die für einen effizienten Betrieb erforderlich sind.
- Ihr Spiel verfolgt möglicherweise das Ziel, mindestens einmal im Monat neue Inhaltsupdates zu veröffentlichen, ohne dass es während der Veröffentlichung zu Unterbrechungen kommt. Diese

Informationen helfen dir dabei, deine Strategie für die Bereitstellung von Releases zu definieren und die Planung erforderlicher Wartungsarbeiten zu koordinieren, die zu anderen Zeiten im Monat zu Ausfallzeiten führen können, und tragen so zu deiner Verfügbarkeits-SLA bei.

Struktur des Kontos

GAMEOPS02: Wie strukturierst du deine Spielumgebungen AWS-Konten für das Hosten?

Implementieren Sie eine Strategie für mehrere Konten, um verschiedene Spielumgebungen zu isolieren und die Sicherheit, betriebliche Effizienz und Skalierbarkeit zu verbessern. Wird verwendet AWS Organizations , um Kontohierarchien zu verwalten, Guardrails auf Konten anzuwenden und Tag-Richtlinien und Tagging für bereitgestellte Ressourcen durchzusetzen.

Best Practices

- [GAMEOPS02-BP01 Verwenden Sie eine Strategie für mehrere Konten, um verschiedene Spiele und Anwendungen in ihren eigenen Konten zu isolieren](#)
- [GAMEOPS02-BP02 Organisieren Sie Infrastrukturressourcen mithilfe von Ressourcen-Tagging](#)

GAMEOPS02-BP01 Verwenden Sie eine Strategie für mehrere Konten, um verschiedene Spiele und Anwendungen in ihren eigenen Konten zu isolieren

Entwerfen Sie eine Kontostruktur, die die Bereitstellung der Infrastruktur so steuert, dass sie den Sicherheits-, Isolations- und Betriebsanforderungen der jeweiligen Umgebung entspricht. Die Isolierung der Umgebung durch Beschränkung des Zugriffs auf die Umgebung und die Nutzung nur der erforderlichen AWS Dienste in diesen Umgebungen ist unerlässlich. Produktionsumgebungen sind gesperrt, während Entwicklungs- und Testumgebungen nachsichtig sind, um Experimente zu ermöglichen. Es wird dringend empfohlen, die wichtigsten Subsysteme in jeder Umgebung weiter zu isolieren und gemeinsame Dienste, die von mehreren Umgebungen genutzt werden, um sie unabhängig voneinander zu hosten und zu verwalten. AWS-Konten

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Verfolgen Sie eine Strategie für mehrere Konten, AWS indem Sie die verschiedenen Umgebungen (wie Entwicklung, Test, Staging, Produktion und gemeinsam genutzte Dienste) für einzelne Benutzer isolieren AWS-Konten, wodurch das Ausmaß der Vorfälle reduziert wird. Erwägen Sie AWS Organizations , Ihre Hierarchie zentral zu verwalten, um die Abläufe weiter AWS-Konten zu vereinfachen, und definieren und wenden Sie Richtlinien auf Kontoebene und Ebene der Organisationseinheiten (OU) selektiv an. Durch die Gestaltung einer geeigneten Organisationseinheit und AWS-Konto Struktur, die auf Ihre Anforderungen in Bezug auf Entwicklungs- und Produktionsabläufe abgestimmt sind, können Sie Ihre Kosten optimieren und die Skalierbarkeit verbessern.

- Verfolgen Sie eine Strategie für mehrere Konten: Isolieren Sie Umgebungen, um die Häufigkeit von Zwischenfällen zu verringern und den Betrieb zu vereinfachen.
- Verwendung AWS Organizations: Konten hierarchisch verwalten, Richtlinien anwenden und zentralisierte Verwaltung ermöglichen.
- Plan für Skalierbarkeit: Entwerfen Sie detaillierte Kontostrukturen und implementieren Sie Maßnahmen zur Kosteneinsparung für future Wachstum.

Implementierungsschritte

Ein in installiertes Spielsystem AWS sollte mehrere Konten verwenden, die logisch organisiert sind, um eine angemessene Isolierung zu gewährleisten. Dadurch wird der Explosionsradius von Problemen reduziert und der Betrieb vereinfacht, wenn Ihre Spielinfrastruktur skaliert. AWS-Konten Diese Host-Spiele-Infrastruktur ist in der Regel in die folgenden logischen Umgebungen unterteilt:

- Spieleentwicklungsumgebungen werden von Entwicklern für die Entwicklung der Software und Systeme für das Spiel verwendet.
- Test- oder Qualitätssicherungsumgebungen (QA) werden zur Durchführung von Integrationstests, manueller Qualitätssicherung und anderen automatisierten Tests verwendet, die durchgeführt werden müssen.
- Staging- oder Vorproduktionsumgebungen werden für das Hosten der fertigen Software verwendet, sodass Last- und Rauchtest vor dem Produktionsstart durchgeführt werden können.
- Live- oder Produktionsumgebungen werden für das Hosten der Live-Software und der Infrastruktur sowie für den Produktionsdatenverkehr der Spieler verwendet.

- Umgebungen mit gemeinsam genutzten Diensten oder Tools bieten Zugriff auf gemeinsame Systeme, Software und Tools, die von vielen verschiedenen Teams verwendet werden. Beispielsweise können ein zentrales, selbst gehostetes Quellcodeverwaltungs-Repository und eine Spiele-Build-Farm in einem Shared Services-Konto gehostet werden.
- Sicherheitsumgebungen werden für die Konsolidierung zentralisierter Protokolle und Sicherheitstechnologien verwendet, die von Teams verwendet werden, die sich auf Cloud-Sicherheit konzentrieren.

Wenn die Spielinfrastruktur aktiviert ist AWS, wird empfohlen, separate Konten für jede Spielumgebung (Entwicklung, Testen, Staging und Produktion) sowie Konten für Sicherheit, Protokollierung und zentrale gemeinsame Dienste einzurichten.

In der Regel können kleinere Spieleentwicklungsstudios, die eine begrenzte Anzahl von Infrastrukturressourcen verwalten, normalerweise einige hundert Server oder weniger, AWS-Konto für jede dieser Umgebungen eines erstellen (z. B. ein Produktionskonto, ein Entwicklungskonto und ein Staging-Konto). Da Ihre Spielinfrastruktur oder Teamgröße jedoch im Laufe der Zeit wächst, lässt sich dieses vereinfachte Modell möglicherweise nicht gut skalieren.

Beachten Sie bei der Einrichtung dieser Umgebungen, dass sich viele AWS Dienste [Service Quotas](#) auf Ressourcen- und API-Ebene für ein gesamtes Konto innerhalb einer bestimmten Region teilen. Dies muss bei der Entscheidung, wie Konten logisch organisiert werden sollen, berücksichtigt werden. AWS-Konten entstehen nur Kosten für die Nutzung der in ihnen bereitgestellten Dienste. Auf diese Weise können Sie Ressourcenkonflikte und Servicequotas effektiv reduzieren, insbesondere wenn Ihr Spiel wächst und immer mehr Entwickler Zugriff auf die Erstellung und Verwaltung von Ressourcen benötigen.

Aufgrund unserer Erfahrung in der Zusammenarbeit mit größeren Spieleentwicklungsstudios, die in der Regel Tausende von Servern betreiben und Hunderte von Entwicklern auf Ressourcen zugreifen, empfehlen wir Ihnen, eine detailliertere Kontostruktur zu entwerfen, bei der einzelne Anwendungen, die Ihr Spiel unterstützen, ihre eigenen Entwicklungs-, Test-, Staging- und Produktionskonten haben. Da es aufgrund der Komplexität bei der Planung und Migration von Live-Systemen schwierig und zeitaufwändig ist, Ihre AWS Multi-Account-Strategie nach dem Start Ihres Spiels neu zu gestalten, sollten Sie bei der Festlegung der richtigen Multi-Account-Struktur Ihre future Skalierungsanforderungen berücksichtigen.

[Sie können AWS Organizationsdamit eine Hierarchie und Gruppierung von Organisationseinheiten \(OUs\) einrichten und definieren AWS-Konten, um mithilfe von Dienststeuerungsrichtlinien \(\) gemeinsame Richtlinien auf OU-Ebene auf sie anzuwenden.](#) SCPs AWS Organizations verwaltet und

steuert Ihre Umgebung zentral, während Sie wachsen und Ihre Ressourcen skalieren. Sie können programmgesteuert neue Konten erstellen und Ressourcen zuweisen, Konten gruppieren, um Ihre Workflows zu organisieren, Richtlinien auf Konten oder Gruppen zur Verwaltung anwenden und die Abrechnung vereinfachen, indem Sie eine einzige Zahlungsmethode für Ihre Konten verwenden. Darüber hinaus ist Organizations in andere Dienste integriert, sodass Sie zentrale Konfigurationen, Sicherheitsmechanismen, Prüfanforderungen und die gemeinsame Nutzung von Ressourcen zwischen Konten in Ihrer Organisation definieren können.

[AWS Control Tower](#) bietet eine einfache Möglichkeit, eine sichere Umgebung mit mehreren Konten einzurichten und zu verwalten, die als landing zone bezeichnet wird. Control Tower schafft Ihre landing zone mithilfe von AWS Organizations fortlaufender Kontoverwaltung und Governance sowie Best Practices für die Implementierung, die auf unserer AWS Erfahrung bei der Zusammenarbeit mit Tausenden von Kunden bei der Umstellung auf die Cloud basieren. [AWS Config](#), [AWS Trusted Advisor](#), und [AWS Security Hub CSPM](#) sind Dienste, die einen aggregierten oder zentralen Überblick über die Hygiene Ihres Kontos bieten.

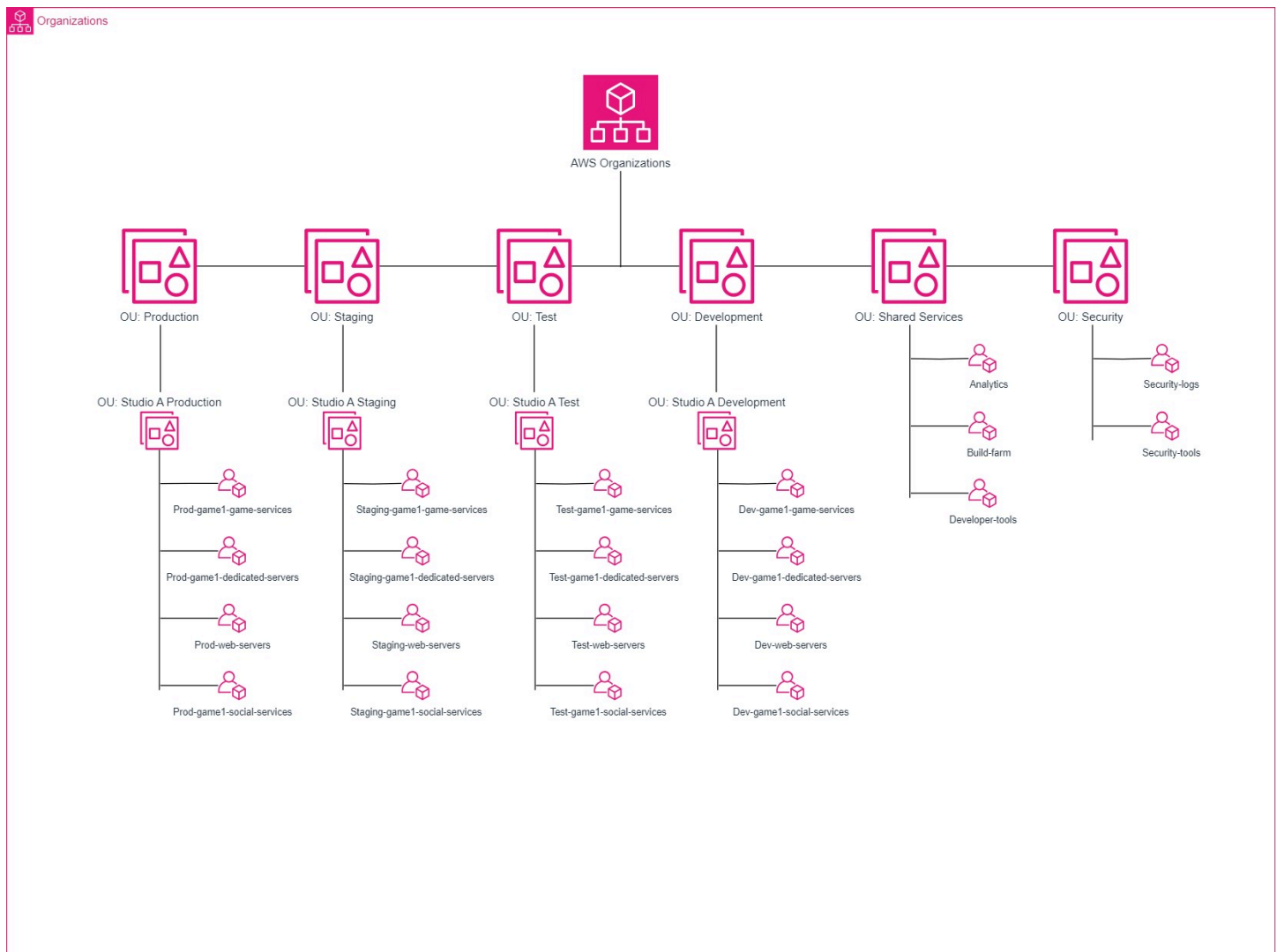
Diese Isolierung hilft dir dabei, benutzerdefinierte oder individuelle Berechtigungen und Schutzmaßnahmen für jede Spielumgebung einzurichten. Produktionskonten sollten über die erforderlichen Schutzmaßnahmen, Zugriffsbeschränkungen, Überwachungs- und Warnmeldungen sowie Sicherheitstools verfügen, während für Konten außerhalb der Produktion möglicherweise nicht die gleichen Schutzmaßnahmen und Berechtigungen erforderlich sind. Umgebungen außerhalb der Produktion können automatisiert werden, um Ressourcen nach Geschäftsschluss herunterzufahren und Kosten zu sparen. Die Trennung der Konten auf dieser Granularitätsebene macht es einfach, die Infrastrukturkosten für jede Umgebung, die ein Spiel unterstützt, zu überwachen.

Im Folgenden finden Sie ein Beispiel für eine Struktur mit mehreren Konten für ein Spieleunternehmen, das Organisationseinheiten (OUs) verwendet, um sich logisch AWS-Konten in separate Umgebungen AWS Organizations und Studios zu gruppieren. In diesem Beispiel OUs werden Accounts anhand ihrer Umgebung und dann anhand des Studios, das die Umgebung betreibt, zusammengefaßt. Dies zeigt, wie Sie eine verschachtelte Hierarchie erstellen können, sodass separate Anwendungen und Spiele in ihren eigenen Konten in ihrer Umgebung (dargestellt als OUs) bereitgestellt werden können. Dies kann nützlich sein, wenn Sie mehrere Spiele entwickeln und betreiben. Weitere Strategien, die Sie für die Organisation Ihrer Multi-Account-Strategie in Betracht ziehen können, finden Sie in der Dokumentation und den Whitepapers im Abschnitt Ressourcen dieser Säule.

Basierend auf den obigen Ausführungen geht das folgende Beispieldiagramm von einem Spielestudio (Organisation) aus, dessen Entwicklungspipeline aus vier Phasen (Entwicklung, Testen, Staging

und Produktion) besteht. Für ein bestimmtes Spiel (Spiel1) verfügt jede der Umgebungen (OU) über eigene AWS-Konten Spieledienste, spezielle Spieleserver, soziale Dienste und Webserver. Die Ressourcen, die in den einzelnen Subsystemen ausgeführt werden, AWS-Konto sind für die jeweiligen Subsysteme relevant. In der Regel würde jedes einzelne Spiel, das diese Art von Entwicklungspipeline verwendet, diese oder eine ähnliche Struktur für sich replizieren. AWS-Konten

Zusätzlich zu diesen spielorientierten Umgebungen OUs gibt es auch die OU für gemeinsame Dienste und die Sicherheits-OU. Diese OUs sollten unternehmensweit gelten, nicht für jedes einzelne Spiel. Auf diese Weise würden die Spiele die gemeinsam genutzten Dienste für Entwicklungstools, Daten und Analysen nutzen, wie in diesem Beispiel. Senden Sie dann Anwendungs- und Systemprotokolle an die AWS-Konto Einrichtung für Protokolle in der Sicherheits-OU.



Beispiel für eine Kontostruktur für Spielumgebungen

GAMEOPS02-BP02 Organisieren Sie Infrastrukturrressourcen mithilfe von Ressourcen-Tagging

Um Ihre [Infrastrukturrressourcen effektiv zu verwalten und nachzuverfolgen AWS](#), sollten Sie [die Ressourcen entsprechend kennzeichnen und gruppieren](#), um den Eigentümer, das Projekt, die Anwendung, die Kostenstelle und andere Daten jeder Ressource zu identifizieren. Mit Tags versehene Ressourcen können mithilfe von [Ressourcengruppen](#) gruppiert werden, was die betriebliche Unterstützung erleichtert.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Definieren Sie [Tagging-Richtlinien](#). Zu den typischen Strategien gehören Ressourcen-Tags zur Identifizierung des Eigentümers der Ressource, z. B. Teamname oder Einzelname, Name des Spiels, Anwendung oder Projekts, Studioname, Umgebung (wie Entwicklung oder Produktion) und Rolle der Ressource (z. B. Datenbankserver, Webserver, dedizierter Spieleserver, App-Server oder Cacheserver). Sie können weitere Tags hinzufügen, um Ihren Geschäfts- und IT-Anforderungen gerecht zu werden. [AWS Config](#) kann auch eine [Tagging-Richtlinie](#) bei der Erstellung und Aktualisierung von Ressourcen durchsetzen. Tags und Ressourcengruppen sind über die Operationen AWS-Managementkonsole AWS CLI, die und die API verfügbar.

Implementierungsschritte

- Taggen Sie Ressourcen, um deren Eigentümer, Projekt, App, Kostenstelle und andere relevante Daten zu identifizieren.
- Implementieren Sie Tagging-Richtlinien, einschließlich Tags für Eigentümer, Projekt, Studio, Umgebung und Ressourcenrolle.
- Wird verwendet AWS Config , um Tagging-Richtlinien durchzusetzen und Tags über AWS-Managementkonsole CLI und API zu verwalten.

Bereitstellungen von Spielen

GAMEOPS03: Wie verwaltet man Spielbereitstellungen?

Verwalte Spielbereitstellungen, indem du wiederverwendete Komponenten gründlich validierst, regelmäßige Performance-Engineerings durchführst und während des gesamten Entwicklungszyklus regelmäßige Belastungstests durchführst.

Best Practices

- [GAMEOPS03-BP01 Validiere und teste deine bestehenden Kernspielsysteme und -infrastrukturen, bevor du sie in deinem Spiel wiederverwendest](#)
- [GAMEOPS03-BP02 Führen Sie vor jeder Veröffentlichung \(oder zumindest bei Hauptversionen\) Performance Engineering durch](#)
- [GAMEOPS03-BP03 Frühzeitiger und häufiger Lasttest](#)
- [GAMEOPS03-BP04 Verfolge eine Einsatzstrategie, die die Auswirkungen auf die Spieler minimiert](#)
- [GAMEOPS03-BP05 Zur Unterstützung von Spitzenanforderungen ist eine vorab skalierbare Infrastruktur erforderlich](#)

GAMEOPS03-BP01 Validiere und teste deine bestehenden Kernspielsysteme und -infrastrukturen, bevor du sie in deinem Spiel wiederverwendest

Organizations neigen dazu, vorhandene Komponenten und Quellcode aus früheren Spielen wiederzuverwenden, um Entwicklungszeit und -kosten zu sparen. Diese älteren Komponenten und der Code werden möglicherweise keiner gründlichen Überprüfung oder detaillierten Integrationstests unterzogen, sondern verlassen sich stattdessen auf ihre bisherige Leistung.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Die Wiederverwendung trägt zwar zur Steigerung der Produktivität bei, kann aber auch das Risiko mit sich bringen, dass frühere Leistungs- und Stabilitätsprobleme in einem neuen Projekt erneut auftreten. Daher sollten bei der Wiederverwendung vorhandener Komponenten und Quellcodes aus früheren Spielen robuste Tests durchgeführt werden.

Implementierungsschritte

- Identifizieren Sie wiederverwendeten Code und Komponenten: Katalogisieren Sie den Quellcode, die Bibliotheken und Komponenten, die aus früheren Spielen wiederverwendet wurden. Unterscheide klar zwischen aktiv gepflegtem und veraltetem Code

- Dokumentieren Sie das ursprüngliche Verhalten und bekannte Probleme: Notieren Sie die ursprünglichen Leistungsmerkmale, Funktionseinschränkungen und bekannte Fehler oder Produktionsvorfälle im Zusammenhang mit den wiederverwendeten Komponenten.
- Führen Sie eine gründliche Überprüfung des Codes durch: Führen Sie eine detaillierte technische Überprüfung der wiederverwendeten Komponenten durch, insbesondere der Komponenten, bei denen in der Vergangenheit Probleme aufgetreten sind oder die schlecht dokumentiert sind.
- Ersetzen oder überarbeiten Sie ältere Komponenten mit hohem Risiko: Priorisieren Sie den Austausch oder die Aktualisierung älterer Komponenten, bei denen in der Vergangenheit Probleme aufgetreten sind oder die nicht mehr gewartet werden können, anstatt sich auf Behelfslösungen in der Produktion zu verlassen.
- Führen Sie Integrations- und Kompatibilitätstests durch: Validieren Sie die wiederverwendeten Komponenten im Kontext der Systeme des neuen Spiels. Stellen Sie sicher, dass sie ordnungsgemäß mit neuen Modulen, Tools und APIs interagieren.

GAMEOPS03-BP02 Führen Sie vor jeder Veröffentlichung (oder zumindest bei Hauptversionen) Performance Engineering durch

Performance Engineering ist der Prozess der Überwachung mehrerer wichtiger Betriebskennzahlen einer App, um Optimierungsmöglichkeiten zu ermitteln, mit denen die Leistung der Anwendung weiter verbessert werden kann. Dies ist ein iterativer Prozess, der mit dem Testen beginnt, gefolgt von der Optimierung des Codes, seiner Abhängigkeiten, der zugehörigen Prozesse, des Host-Betriebssystems und der zugrunde liegenden Infrastruktur.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Um die Leistung der App eingehender zu analysieren, integrieren Sie ein Tool zur Überwachung der Anwendungsleistung (APM) oder zum Debuggen in den App-Code, mit dem Probleme isoliert und die Zeit für die Fehlerbehebung reduziert werden können, indem das Verhalten des Programms im Hinblick auf Anomalien in allen Abläufen der App verfolgt wird. APM-Tools sind auch in der Lage, langsam funktionierende Methoden und externe Operationen zu identifizieren.

[AWS X-Ray](#) unterstützt Entwickler bei ihren Performance-Engineering-Aktivitäten wie der Identifizierung von Leistungsengpässen und der Analyse und Debugging von Produktionsfehlern. Sie können X-Ray verwenden, um die Leistung Ihrer Anwendung und der zugrunde liegenden Dienste zu verstehen und die Hauptursache von Leistungsproblemen und Fehlern zu identifizieren und zu

beheben. In zahlreichen Auslastungstestrunden, in denen die Anwendung und ihre Infrastruktur nach und nach mit synthetischem Spielerverkehr belastet werden, werden verschiedene Systemengpässe, Anwendungsfehler, Ausnahmen, Betriebssystemprobleme und andere Probleme identifiziert, die bei anderen QA-Tests möglicherweise nicht gefunden wurden.

Verwenden Sie für kritische Ereignisse wie Spieleinführungen, Inhaltsveröffentlichungen, Werbeaktionen und wichtige Ereignisse im Spiel Countdown. [AWS Countdown](#) bietet Implementierungsleitfäden auf der Grundlage von Playbooks, die von Spieleexperten erstellt wurden, um die Betriebsbereitschaft zu überprüfen, potenzielle Risiken zu minimieren und den Kapazitätsbedarf zu planen. AWS Countdown bietet auch eine [Premium-Supportoption](#), die erweiterten Support und Optionen wie Techniker zur Optimierung Ihrer Infrastruktur bietet.

Implementierungsschritte

- Performance Engineering umfasst die Bewertung und Überwachung wichtiger Betriebskennzahlen, um sicherzustellen, dass der Code, die Prozesse, das Betriebssystem und die Infrastruktur Ihrer Anwendung wie erwartet funktionieren. Die Überprüfung vor der Produktion hilft auch dabei, die Ausgangsleistung auf verschiedenen Ebenen der simulierten Nutzung zu definieren.
- Ermitteln und verfolgen Sie wichtige Kennzahlen wie Auslastung, Services, I/O, Prozesse usw. mithilfe von Systemtools wie sar, top, vmstat, sysstat, netstat und Performance Monitor.
- Verfolgen Sie die Leistung und das Verhalten Ihrer Anwendung mithilfe von APM-Tools, um beispielsweise Probleme AWS X-Ray zu isolieren, Engpässe zu identifizieren und Produktionsfehler zu debuggen.
- Abonnieren Sie AWS Countdown (IEM) für kritische Ereignisse wie Spieleinführungen, um architektonische und betriebliche Beratung, operativen Support auf Abruf sowie um Risiken zu identifizieren und Abhilfemaßnahmen zu planen.

GAMEOPS03-BP03 Frühzeitiger und häufiger Lasttest

Beim Auslastungstest wird der reale Verkehr auf einem System simuliert, um dessen Zuverlässigkeit und Leistung zu bewerten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Auslastungstests sind ein wichtiger Faktor bei der Entwicklung einer Leistungsbasis für Ihre Ressourcen und beim Verständnis der Kapazität Ihres Systems. Dies kann als

Grundlage für Finanzprognosen, Architekturdesign, Ressourcenzuweisung, automatisierte Skalierungskonfigurationen und Aktivitäten vor der Skalierung nach dem Start dienen. Weitere Vorteile:

- **Optimierte Infrastruktur:** Ressourcen sind möglicherweise über- oder unterversorgt. Wenn Sie wissen, welche Ressourcen benötigt werden, können Sie die Kosten senken und die zu verwaltende Infrastruktur verringern.
- **Bereitschaft zur Skalierbarkeit:** Bestimmte Mechanismen und Funktionen können dazu führen, dass Benutzer schnell in ein Spiel einsteigen. Zu wissen, wann und wie skaliert werden muss, kann den Unterschied zwischen der angemessenen Erfüllung der gestiegenen Nachfrage und dem Verlust von Spielern ausmachen. Verwenden Sie die Ergebnisse von Auslastungstests, um Runbooks mit Systemschwellenwerten, Warnpunkten und kritischen Warnpunkten auf unterschiedlichen Skalierungsebenen vorzubereiten.
- **Höherer Qualitätscode:** Probleme wie übermäßiger Crosstalk zwischen Diensten, Datenbankabrufe ohne Batch, ineffiziente Algorithmen, Speicherlecks und Probleme mit der Beeinträchtigung von Diensten lassen sich manchmal leichter in größerem Umfang identifizieren.
- **Verhaltensvalidierung:** Wenn Sie verschiedene Arten von Fehlern in Ihre Tests einbeziehen, können Sie das erwartete Verhalten des Systems validieren oder Probleme bei der Fehlerbehandlung aufdecken, die behoben werden müssen.

Idealerweise sollten Entwickler Lasttests an mehreren Stellen des Entwicklungsprozesses durchführen, da jeder unterschiedliche Vorteile mit sich bringen kann: Schon früh leiten sie Architekturentscheidungen und Refactoring-Bemühungen, während es billiger und einfacher ist, Änderungen vorzunehmen. Am Ende jedes Sprints oder jeder Iteration validieren sie die Leistung der Anwendung anhand der neuesten Features und Funktionen.

Vor der Bereitstellung in der Produktion wird durch groß angelegte Lasttests, bei denen zu erwartende reale Nutzungsmuster simuliert werden, bestätigt, dass das System in der Lage ist, die Produktionslast zu bewältigen. Nach der Bereitstellung überwachen regelmäßige Lasttests die Leistung des Systems und identifizieren Änderungen oder Engpässe, die im Laufe der Zeit auftreten können.

Um den Spielerverkehr zu simulieren, benötigen Sie einfache Clients oder Bots, die die Abläufe der Spielclients emulieren und Transaktionen mit dem Spiel-Backend durchführen, um das reale Spielerverhalten zu simulieren. Diese Daten werden im Allgemeinen anhand von Spielprotokollen und Daten erfasst, die durch von Menschen gesteuerte QA-Tests generiert wurden, sowie durch Alpha-

oder Betatests in begrenztem Umfang unter realen Bedingungen, bei denen echte Spieler eingeladen werden, eine Early-Access-Version des Spiels zu spielen.

Es ist wichtig, das Verhalten des Systems in einem Runbook aufzuzeichnen, um mögliche Fehler in der future beheben zu können und Leistungskennzahlen beizubehalten, mit denen future Lasttests verglichen werden können. Es wird außerdem empfohlen, das Spiel während des Auslastungstests von menschlichem QA-Personal testen zu lassen, da sie möglicherweise Probleme entdecken, die Bots nicht identifizieren können und die Metriken nicht widerspiegeln.

AWS Der [Fault Injection Service](#) ist ein vollständig verwalteter Service für die Durchführung von Fault-Injection-Experimenten, mit denen die Leistung, Beobachtbarkeit und Stabilität einer Anwendung auf einfache Weise verbessert werden kann. Fault-Injection-Experimente werden im Bereich Chaos Engineering eingesetzt. Dabei handelt es sich um eine Praxis, bei der eine Anwendung in Test- oder Produktionsumgebungen durch störende Ereignisse wie einen plötzlichen Anstieg des CPU- oder Speicherverbrauchs belastet wird, beobachtet wird, wie das System reagiert, und Verbesserungen implementiert werden. Experimente zur Fehlerinjektion helfen Teams dabei, die realen Bedingungen zu schaffen, die erforderlich sind, um versteckte Fehler aufzudecken, blinde Flecken zu überwachen und Leistungsengpässe zu überwachen, die in verteilten Systemen schwer zu finden sind.

Implementierungsschritte

- Richten Sie mithilfe von [Guidance for Kubernetes-Bases Game Load Testing eine Umgebung für verteilte Lasttests](#) ein.
- Passen Sie Locust Control und Worker Pods innerhalb des EKS-Clusters mithilfe der bereitgestellten Bereitstellungsdateien an und implementieren Sie sie, um eine skalierbare und verwaltbare Lastgenerierung zu ermöglichen.
- Erfassen Sie das Systemverhalten und die Messwerte während des Auslastungstests in einem Runbook, um Sie bei der future Fehlerbehebung zu unterstützen und Leistungsbasislinien festzulegen.
- Verwenden Sie Fault-Injection-Experimente, um Störungen in der realen Welt zu simulieren und versteckte Probleme in Bezug auf Systemleistung, Beobachtbarkeit und Ausfallsicherheit aufzudecken.

GAMEOPS03-BP04 Verfolge eine Einsatzstrategie, die die Auswirkungen auf die Spieler minimiert

Integrieren Sie eine Implementierungsstrategie für Ihre Spielesoftware und -infrastruktur, die die Ausfallzeiten minimiert, die Spieler vom Spiel fernhalten. Bei bestimmten Arten von Updates müssen möglicherweise neue Updates für den Spielclient installiert werden. Gestalten Sie das Spiel jedoch so, dass Ausfallzeiten während der Bereitstellung minimiert oder vermieden werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Einer der wichtigsten Schritte, die Sie bei der Entwicklung einer Strategie für die Bereitstellung von Spielen berücksichtigen sollten, besteht darin, festzulegen, wie Ihre Spieleinfrastruktur verwaltet werden soll. Verwalte deine Spieleinfrastruktur mit einem Infrastructure-as-Code-Tool (IaC) wie [AWS CloudFormation](#) [Terraform von Hashicorp](#), um menschliche Fehler bei der Vorbereitung der [Umgebung](#) zu vermeiden. Infrastrukturvorlagen können in automatisierten Pipelines bereitgestellt und getestet werden, wodurch eine einheitliche Konfiguration der verschiedenen Spielumgebungen gewährleistet wird.

Es gibt verschiedene Bereitstellungsstrategien, die für ein Spiel verwendet werden können:

Fortlaufende Substitution

Das Hauptziel einer fortlaufenden Substitution für den Einsatz besteht darin, die Veröffentlichung durchzuführen, ohne das Spiel zu beenden und ohne die Spieler zu beeinträchtigen. Es ist wichtig, dass das Upgrade oder die vorzunehmenden Änderungen abwärtskompatibel sind und neben den vorherigen Versionen des Systems funktionieren.

Bei dieser Bereitstellung werden die Serverinstanzen schrittweise durch Instanzen ersetzt (ersetzt oder eingeführt), auf denen die aktualisierte Version ausgeführt wird. Diese fortlaufende Substitution kann auf verschiedene Arten durchgeführt werden. Um beispielsweise fortlaufende Updates für eine Flotte von dedizierten Spieleservern zu implementieren, besteht ein typischer Ansatz darin, eine neue Auto Scaling Group von EC2 Instances zu erstellen, die die auf ihnen bereitgestellte neue Spielserver-Build-Version enthalten, und die Spieler dann schrittweise zu Spielsitzungen weiterzuleiten, die auf dieser neuen Serverflotte gehostet werden. Wenn es ein zugehöriges Spielclient-Update gibt, das als Voraussetzung für die Verwendung des neuen Spielserver-Builds erforderlich ist, müssen Sie eine Validierungsprüfung einschließen, um sicherzustellen, dass

nur Spieler, auf denen dieses neue Spielclient-Update installiert ist, zu diesen Spielsitzungen weitergeleitet werden.

Serverflotten (z. B. EC2 Auto Scaling Scaling-Gruppen), die die alte Build-Version des Spieleservers enthalten, werden erst dann außer Betrieb genommen, wenn ihnen die aktiven Spiellersitzungen ordnungsgemäß entzogen wurden, in der Regel durch die Einrichtung individueller Servermetriken, die es den Spielbetriebsteams ermöglichen, diesen Prozess zu automatisieren. Um den Umfang der Infrastruktur und die Zeit für die Durchführung einer fortlaufenden Bereitstellung zu reduzieren, kann alternativ ein alternativer Ansatz verfolgt werden, bei dem bestehende Produktionsinstanzen außer Betrieb genommen, mit dem neuen Spieleserver-Build aktualisiert und dann wieder in die Produktionsflotte aufgenommen werden. Dieser Ansatz reduziert den Umfang der benötigten Infrastruktur, erhöht aber auch das Risiko, da die Anzahl der verfügbaren Live-Spieleserver für Spieler reduziert wird, wenn Server ersetzt werden.

Dieses Modell kann auch für die fortlaufende Bereitstellung von Backend-Diensten wie Datenbanken, Caches und Anwendungsservern verwendet werden, die kein Gameplay hosten. Solange diese Dienste auf hochverfügbare Weise mit mehreren Clusterinstanzen bereitgestellt werden, sollte die Komplexität der Bereitstellungen für diese Dienste geringer sein als die von Bereitstellungen auf dedizierten Spieleservern.

Einsatz in Blau/Grün

Das Hauptziel einer blue/green Bereitstellung in einem Spiel besteht darin, Ausfallzeiten zu minimieren und gleichzeitig ein sicheres Rollback zur vorherigen Bereitstellung zu ermöglichen, falls Probleme festgestellt werden. Es eignet sich für Bereitstellungen, bei denen zwei Versionen des Spiel-Backends kompatibel sind und Spielern gleichzeitig zur Verfügung stehen.

In der blue/green Bereitstellungsstrategie werden zwei identische Umgebungen (blau und grün) eingerichtet. Die bestehende Spielversion ist blau gekennzeichnet, während die neue Spielversion, die das Einsatzziel ist, grün gekennzeichnet ist. Wenn die grüne Umgebung für die Migration bereit ist, können Sie Ihre Routing-Ebene so konfigurieren, dass der Datenverkehr auf die grüne Umgebung umgestellt wird und gleichzeitig die alte Umgebung (blau) verfügbar bleibt, falls ein Failback erforderlich ist. In diesem Szenario erfordern die Routing-Updates möglicherweise eine Aktualisierung des Matchmaking-Service, um ihn so zu konfigurieren, dass er mit dem Senden von Spielsitzungen an die neue Flotte beginnt, oder im Fall von Backend-Diensten für Spiele könnte dies die Aktualisierung der DNS-Einträge in Amazon Route 53 für Ihren Service oder die [Verschiebung der Gewichte des Application Load Balancers](#) sein, um Traffic an Ihre neue Zielgruppe zu senden.

Einer der Nachteile der blue/green Bereitstellungsstrategie sind die mit der Standby-Umgebung verbundenen Kosten, die auf die zusätzliche Infrastruktur zurückzuführen sind, die während der Bereitstellung erforderlich ist. Eine Möglichkeit, diese zusätzlichen Infrastrukturkosten zu verringern, besteht darin, die Einführung einer blue/green Bereitstellungsvariante in Betracht zu ziehen, bei der neue Spielesoftware auf denselben Servern bereitgestellt wird, die bereits in der Produktion eingesetzt werden. In diesem Szenario kann neben dem bestehenden Blue-Server-Prozess auch ein neuer umweltfreundlicher Serverprozess mit der neuen Software gestartet werden, wobei die Umstellung zwischen Serverprozessen und nicht zwischen separaten physischen Infrastrukturen erfolgt. Dieser Ansatz kann auch die Bereitstellung von Spielen in einem großen Teil der Infrastruktur beschleunigen, da nicht mehr auf den Start neuer Server in der Cloud gewartet werden muss. Bewährte Methoden für diesen Bereitstellungsansatz finden Sie unter [Blue/Green Deployments](#) on AWS

Bereitstellung auf Canary

Die Bereitstellung auf Canary ist für Spieleentwickler nützlich, da die Strategie angewendet werden kann, um eine frühe Alpha- oder Betaversion eines Spiels oder eine Spielfunktion wie einen neuen Spielmodus, eine neue Karte oder eine Herausforderung für eine eingeschränkte oder kleine Gruppe von Spielern in der Produktion zu veröffentlichen. Ein solcher Einsatz wird als Kanarienvogel bezeichnet. Die Version kann zusätzliche Nachverfolgungs- und Berichtsfunktionen beinhalten. Wenn also echte Spieler dieses Spiel oder diese Funktion spielen, werden ihre Spieltelemetriedaten erfasst und auf Anomalien und Probleme analysiert.

Bei neuen Funktionen werden die Spieler nicht regelmäßig darüber informiert, und die Spieltelemetrie ist die wichtigste Quelle, anhand derer festgestellt wird, ob bei Spielern Probleme auftreten und die Veröffentlichung rückgängig gemacht werden sollte. Wenn keine nennenswerten Probleme festgestellt werden, kann die Funktion gleichzeitig weiteren Spielern zur Verfügung gestellt werden, um zusätzliche Daten zu erhalten. Wenn die Spieler benachrichtigt werden, können sie gebeten werden, regelmäßig Feedback zu ihren Erfahrungen zu geben. Solche Testaktivitäten würden idealerweise von einem Live-Operationsteam koordiniert.

Als Strategie kann der Einsatz von Canary auch für Standardversionen verwendet werden, um den Spielern schrittweise ein neues Feature zur Verfügung zu stellen. Ein potenzieller Vorteil gegenüber der blue/green Standardumgebung besteht darin, dass keine vollständige zweite Umgebung erforderlich ist. Die Kapazität der neuen herunterskalierten Umgebung bestimmt, wie viele Spieler mit der neuen Funktion vertraut werden sollen. Bevor weitere Spieler hinzugefügt werden, muss die Kapazität entsprechend skaliert werden. Auch wenn davon ausgegangen wird, dass diese maßgeschneiderte blue/green Technik vergleichsweise weniger Kosten wird als die standardmäßige

Blau/Grün-Technik, wird dennoch davon ausgegangen, dass sie Kosten verursacht, die höher sein können als die fortlaufende Substitutionstechnik bei kanarischen Installationen.

Führen Sie nur einen einzigen Canary in einer Produktionsumgebung aus und konzentrieren Sie sich auf dessen Daten und Feedback. Wenn mehrere Canaries eingesetzt werden, erschwert dies die Behebung und Isolierung von Problemen in der Produktion und beeinträchtigt die Qualität der Datensätze und des gesammelten Feedbacks.

Eine Variante des Kanariensystems besteht darin, dass ein oder mehrere Experimente (in der Regel UI-Tests) durch gezielte Bereitstellungen durchgeführt werden, wobei ein Satz der Backend-Server des Spiels eine Version einer Funktion und eine andere Gruppe derselben Größe eine andere Version derselben Funktion bereitstellt. Dafür wird keine zusätzliche oder spezielle Infrastruktur geschaffen, und nur die ausgewählten Backend-Server erhalten diese Updates. Das Ergebnis der Experimente besteht darin, zu beobachten, wie die Spieler auf die einzelnen Versionen derselben Funktion reagieren, festzustellen, ob allgemein Zustimmung oder Abneigung besteht, und zu beobachten, ob Probleme mit der Benutzerfreundlichkeit oder Funktionalität festgestellt wurden. Solche strategischen Experimente werden auch als A/B Tests bezeichnet, und der gesamte Prozess wird als A/B-Testing bezeichnet. Nach Abschluss dieser Experimente werden die erforderlichen Testdaten gesammelt, bevor auf den für die Tests verwendeten Servern zur aktuellen Version des Spiel-Backend-Systems zurückgekehrt wird.

Herkömmliche ältere Bereitstellungen

Bei der herkömmlichen Art der Bereitstellung wird das Spiel während eines geplanten Wartungsfensters heruntergefahren und verbundene Spieler fallen gelassen oder ausgelaugt, bevor die Serverinstanzen im Spiel-Backend mit den neuesten Code-Builds aktualisiert werden. Diese Bereitstellung wirkt sich jedes Mal auf die Spieler aus, wenn sie durchgeführt wird, und die Spieler müssen im Voraus darüber informiert werden. Aus diesem Grund hat dieses Modell die meisten Auswirkungen auf die Spieler und sollte nach Möglichkeit vermieden werden.

Nach der Veröffentlichung des Spielupdates kann das Spiel einem Rauchttest unterzogen werden, bevor das Spiel für die Spieler geöffnet wird, die dann auf die Wiedereröffnung des Spiels warten würden. Dies kann zu einem Anstieg des Traffics führen, wenn Spieler versuchen, sich innerhalb kurzer Zeit einzuloggen und zu spielen. Wenn das Spiel nicht darauf ausgelegt ist, solche Verkehrsspitzen zu bewältigen, können Sie sich daher dafür entscheiden, die Spieler nach und nach stapelweise wieder ins Spiel zu lassen.

Alternativ könnt ihr euch dafür entscheiden, die Infrastruktur übermäßig bereitzustellen, um den anfänglichen Anstieg des Datenverkehrs aufrechtzuerhalten. Sobald sich der Spieldatenverkehr

stabilisiert hat, können die Ressourcen heruntergefahren werden. Falls erforderlich, sollten Sie diese Art der Bereitstellung außerhalb der Spitzenzeiten durchführen, wenn die Spielerzahl am niedrigsten ist. Häufig geplante Wartungsarbeiten sowie längere Wartungsarbeiten bergen naturgemäß das Risiko einer Fluktuation von Spielern und potenziellen Umsatzeinbußen. Spieler erwarten auch nach einer neuen Version Änderungen und können das Vertrauen in das Spiel verlieren, wenn sie nach einer gewissen Zeit der Ausfallzeit zurückkehren.

Implementierungsschritte

- Minimiere Ausfallzeiten: Implementiere Einsatzstrategien, die Ausfallzeiten reduzieren und die Spieler im Spiel halten.
- Infrastruktur als Code (IaC): Verwende Tools wie Terraform AWS CloudFormation oder Terraform, um die Spielinfrastruktur zu verwalten und menschliche Fehler zu reduzieren.
- Einsatzstrategien: Nutze eine oder mehrere Kombinationen aus fortlaufender Substitution, Blau/Grün und Kanariemodus, um reibungslose Updates bereitzustellen und die Auswirkungen auf die Spieler zu reduzieren.

GAMEOPS03-BP05 Zur Unterstützung von Spitzenanforderungen ist eine vorab skalierbare Infrastruktur erforderlich

Skalieren Sie die Infrastruktur im Vorfeld großer Spielevents, um sicherzustellen, dass Sie den plötzlichen Anstieg der Spielernachfrage bewältigen können.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Neben der Veröffentlichung neuer Spiele bieten Live-Spiele in der Regel auch In-Game-Events, Werbeaktionen, neue Inhalte und Saisonveröffentlichungen als Beispiele dafür, wie die Spielerbindung aufrechterhalten und verbessert werden kann. Bei solchen Aktivitäten kommt es während der Dauer des Events oder der Promotion zu einem hohen Besucheraufkommen. Das Unternehmen geht davon aus, seine angestrebten Ziele für das Event zu erreichen oder zu übertreffen, und die Spieleinfrastruktur muss es dabei unterstützen und unterstützen.

Bereiten Sie Ihre Infrastruktur im Voraus vor, um die zu erwartende Spielerbelastung bewältigen zu können, die Sie bei Großveranstaltungen erwarten werden. Zur Vorbereitung sollten sich die Teams für den Spielbetrieb mit den Stakeholdern aus Vertrieb und Marketing abstimmen, um die

prognostizierte Nachfrage zu schätzen, die bei einer bevorstehenden Veranstaltung generiert werden wird. Dabei sollten sie sich die bisherige Anzahl der Spieler, die Kundenbindung und die Verkaufsdaten ansehen. Wenn es sich bei der Veranstaltung um die Markteinführung eines neuen Spiels handelt, sollten die Teams für den Spielbetrieb mit diesen Beteiligten zusammenarbeiten, um realistische Prognosen für den Umfang zu erstellen, den sie erwarten. Es mag zwar schwierig sein, vorherzusagen, wie erfolgreich ein Spiel werden wird, aber es ist wichtig, dass jeder versteht, welche Erwartungen an den Erfolg gestellt werden, damit die Infrastruktur skaliert und getestet werden kann, um diese Ziele zu erreichen.

Viele Spiele entscheiden sich dafür, stufenweise zu starten. Sie beginnen mit einem Soft-Launch, bei dem das Spiel einer kleinen Anzahl von Spielern zugänglich gemacht wird, und dann die Spieler in jeder Phase organisch skalieren, bevor es vollständig veröffentlicht wird. Während der Soft-Launch-Phase sollten Sie Probleme beobachten, identifizieren, nachverfolgen und lösen und gleichzeitig Ihre Prognosen für den öffentlichen Launch verfeinern.

Um die Infrastrukturanforderungen richtig einschätzen zu können, sollten Sie vor der Veröffentlichung des Spiels Daten mithilfe von Last- und Leistungstests sammeln, die an Ihren Spiele-Backends durchgeführt werden, die in der Produktion oder in einer produktionsähnlichen Staging-Umgebung ausgeführt werden. Es sollten mehrere Runden dieser Tests durchgeführt werden, um verschiedene Bedingungen im Spiel zu simulieren und zu überprüfen, ob das Backend der Last unter den meisten Bedingungen standhält.

Um dies zu erreichen, können Entwickler Gameplay-Bots schreiben, die verschiedene Workflows im Spiel durchlaufen und verschiedene Bedingungen emulieren. Bei diesen Tests sollten die verschiedenen Systemschichten des Spiel-Backends untersucht werden, sodass jede Ebene und Komponente getestet und die Details aufgezeichnet werden. Nutze die bei diesen Tests gesammelten Daten, um einen Plan für den Start des Spiels zu erstellen.

Single Points of Failure (SPOF) sollten identifiziert und nach Möglichkeit entfernt werden, indem die Anwendung hochverfügbar und fehlertolerant gemacht wird. Verwenden Sie Auslastungstests zur Identifizierung, SPOFs indem Sie Fehler auf verschiedenen Upstream- und Downstream-Ebenen emulieren und das Verhalten von Spielen und anderen Komponenten überprüfen.

Richten Sie das System so ein, dass es bei Bedarf automatisch skaliert werden kann, und zwar zusammen mit der voraussichtlichen Infrastruktur, die für den Start des Spiels, die Vorbereitung der Veranstaltung im Spiel oder für die Vorbereitung der Werbeaktion benötigt wird. Definieren, konfigurieren und überwachen Sie Schwellenwerte für Skalierungsereignisse, damit das Spiel-Backend skaliert werden kann, um ein hohes Spieleraufkommen aufrechtzuerhalten. Bei variablem

Datenverkehr ist eine Vorabbereitstellung am besten, da möglicherweise nicht genügend Zeit für eine Skalierung zur Verfügung steht. Bei ersten Spielstarts kann eine manuelle Skalierung erforderlich sein, da die Nachfrage schneller als erwartet steigt, als automatisierte Systeme Ressourcen skalieren können.

AWS Aktiviert, sollten Organisationen höhere [Service Quotas](#) für die Dienste beantragen, die sie im Spiel-Backend verwenden. Service Quotas werden für Konten eingerichtet, um zu verhindern, dass Kunden versehentlich mehr Infrastruktur einrichten oder skalieren als beabsichtigt. Wenn ein Spiel, das auf einem Konto ausgeführt wird, die Obergrenze des konfigurierten Dienstkontingents in dieser Region erreicht, drosselt der Dienst die Anfragen, die über das bereitgestellte Kontingent hinausgehen, und erhöht die Anzahl der Bereitstellungen. Drosselungen können zu unbeabsichtigten oder unerwarteten Fehlern führen und das Spielerlebnis beeinträchtigen. Überwache, verfolge und überprüfe regelmäßig die Schwellenwerte für die Dienste, die vom Spiel während der Produktion genutzt werden, um Drosselungen zu vermeiden. Wenn die Nutzung einen akzeptablen Schwellenwert für das Servicekontingent überschreitet, kann eine Erhöhung des Kontingents beantragt werden, indem eine [Support-Anfrage](#) im Console Support Center gestellt wird, nachdem Sie sich bei dem betroffenen Konto angemeldet haben, oder indem Sie die [Support-API](#) verwenden.

[Verwenden AWS Sie Countdown für wichtige Ereignisse wie Spieleinführungen, Inhaltsveröffentlichungen, Werbeaktionen und wichtige Ereignisse im Spiel.](#) Countdown bietet Anleitungen zur Implementierung auf der Grundlage von Playbooks, die von Spieleexperten erstellt wurden, um die Betriebsbereitschaft sicherzustellen, potenzielle Risiken zu minimieren und den Kapazitätsbedarf zu planen. AWS Countdown bietet auch eine [Premium-Supportoption](#), die erweiterten Support und Optionen wie Techniker zur Optimierung Ihrer Infrastruktur bietet.

Wenn Sie ein auf Amazon gehostetes Spiel auf den Markt bringen GameLift, lesen Sie sich zur Vorbereitung die [Checklisten vor der Veröffentlichung](#) durch.

Implementierungsschritte

- Skalieren Sie die Infrastruktur im Voraus: Bereiten Sie die Infrastruktur im Voraus für große Spielevents vor, um dem plötzlichen Anstieg der Spielernachfrage gerecht zu werden.
- Nachfrage abschätzen: Stimmen Sie sich mit Vertrieb und Marketing ab, um die prognostizierte Nachfrage anhand vergangener Spielerdaten und realistischer Prognosen abzuschätzen.
- Auslastungstests und SPOF-Entfernung: Führen Sie mehrere Runden von Lasttests durch, um die Backend-Kapazität zu validieren, einzelne Fehlerquellen zu identifizieren und die automatische Skalierung ordnungsgemäß zu konfigurieren.

Überwachung Health Gesundheitszustands

GAMEOPS04: Wie überwachst du den Zustand des Spiels?

Überwachen Sie den Zustand des Spiels, indem Sie umfassende Instrumente implementieren, um Probleme zu erkennen und nachzuverfolgen, die sich auf die Spieler auswirken. Dazu gehören die Protokollierung von Aktivitäten auf der Clientseite, die Überwachung von Backend-Services und die Fehlerberichterstattung. Verwenden Sie eine Kombination von AWS Tools wie Amazon CloudWatch und AWS X-Ray Lösungen von Drittanbietern, um Probleme schnell zu identifizieren und zu lösen.

Best Practices

- [GAMESOPS04-BP01 Instrumentieren Sie das Spiel, um Probleme zu erkennen und zu überwachen, die sich auf die Spieler auswirken](#)

GAMESOPS04-BP01 Instrumentieren Sie das Spiel, um Probleme zu erkennen und zu überwachen, die sich auf die Spieler auswirken

Neben der Reaktion auf Probleme in sozialen Netzwerken und Spielermeldungen sollten Sie Ihr Spiel auch mit Überwachungslösungen ausstatten, um Probleme, die sich auf Spieler auswirken, zu erkennen und zu untersuchen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Durch umfangreiche Tests können nicht alle Probleme in einem Spiel identifiziert werden. Spiele werden normalerweise mit bekannten Problemen gestartet, die mit der nächsten Version des Spiels schrittweise behoben werden sollen. Bekannte und reproduzierbare Probleme lassen sich leicht beheben und beheben. Um bei der Identifizierung solcher Probleme zu helfen, sollten Spieleclients an verschiedenen strategischen Stellen die Erfassung von Spieleraktivitäten, die Protokollierung von Apps und die Berichterstattung implementieren, um das Backend-Team bei der Identifizierung von Problemen auf Kundenseite zu unterstützen. Die Fähigkeit, solche Probleme frühzeitig zu finden, hilft den Spieleentwicklern, das Problem zu beheben und zu beheben, bevor es sich ausbreitet. Die vom Tracking-Code gemeldeten Daten und Protokolle sollten niemals personenbezogene Daten (PII) enthalten und sie sollten nur spielspezifische Metadaten enthalten, die beim Debuggen helfen.

Implementieren Sie eine Observability-Lösung, um Probleme wie Spielabstürze oder Bugs zu erkennen und darauf zu reagieren. Sie können [Amazon CloudWatch Synthetics](#) verwenden, um Kanarien zu erstellen, die den Zustand Ihrer Backend-Spieledienste für Spieler überwachen können. Sie können Ihre Backend-Services nutzen, [AWS X-Ray](#) um Anfragen über verteilte Dienste hinweg zu verfolgen und Ihre benutzerdefinierten Protokolle und Metriken an [Amazon CloudWatch](#) zu senden.

Lösungen von Drittanbietern wie [Backtrace.io](#) und [Sentry](#) sind beliebte Lösungen für die Fehlerberichterstattung in Spielen. [Lösungen zur Überwachung der Anwendungsleistung \(Application Performance Monitoring, APM\) von Partnern wie New Relic, Splunk, Datadog und Honeycomb.io sind ebenfalls beliebt.](#)

Das Live-Betriebsteam und die Community-Manager des Spiels sollten zusätzlich zu den offiziellen Support-Kanälen auch verschiedene soziale Netzwerke und Kanäle im Auge behalten, um nach Feedback, Beschwerden und Bugreports von Spielern zu suchen. Überprüfe jede spielspezifische Beschwerde und versuche, sie zu reproduzieren, oder schicke sie zur Überprüfung an das QA-Team. Falls es reproduzierbar ist, leiten Sie das Problem an die Spieleentwickler weiter, damit sie es beheben und beheben können, bevor es sich auf die größere Spielerbasis auswirkt.

Implementierungsschritte

- Implementieren Sie Überwachungslösungen: Verwenden Sie Überwachungstools, um Probleme zu erkennen, die sich auf die Spieler auswirken, und reagieren Sie schnell.
- Spieleraktivitäten und -protokolle verfolgen: Instrumentieren Sie Spielclients, um Spieleraktivitäten zu protokollieren und Probleme zu melden, und stellen Sie sicher, dass keine personenbezogenen Daten (PII) enthalten sind.
- Verwenden Sie AWS Tools von Drittanbietern: Verwenden Sie Tools wie CloudWatch X-Ray und Lösungen von Drittanbietern für die Fehlerberichterstattung und Leistungsüberwachung und überwachen Sie soziale Medien auf Spielerfeedback und Fehlerberichte.

Lasttest

GAMEOPS05: Was solltest du beim Auslastungstesten eines Spiels beachten?

Berücksichtigen Sie beim Auslastungstest eines Spiels die geeignete Testphase, die Architektur zur Lastgenerierung und das Testframework, um die Systemleistung und Skalierbarkeit effektiv

bewerten zu können. Wählen Sie die richtige Kombination aus Timing (frühe Entwicklung, Sprints, Vorproduktion oder nach der Bereitstellung), Infrastruktur (, EKSEC2, Fargate oder Lambda) und Testtool (, LocustJMeter, Grafana K6 oder Gatling), um sie an die einzigartigen Eigenschaften und Entwicklungsziele Ihres Spiels anzupassen.

Best Practices

- [GAMEOPS05-BP01 Wählen Sie die richtige Phase, Architektur und das richtige Lasttesting-Framework, um Ihre Ziele zu erreichen](#)

GAMEOPS05-BP01 Wählen Sie die richtige Phase, Architektur und das richtige Lasttesting-Framework, um Ihre Ziele zu erreichen

Die Vorgehensweise beim Auslastungstest eines Spiels kann stark variieren. Dies hängt von vielen Faktoren ab, darunter der Phase des Entwicklungsprozesses, in dem das Spiel durchgeführt wird, der Architektur des lastgenerierenden Systems selbst und der Wahl des Frameworks für Lasttests. Der Zeitpunkt, zu dem die Tests durchgeführt werden, sei es in den frühen Phasen, während iterativer Sprints, vor der Produktionsbereitstellung oder nach der Bereitstellung, bestimmt die Ziele und den Schwerpunkt der Testbemühungen. Verschiedene Designs der lastgenerierenden Infrastruktur haben ihre eigenen Vor- und Nachteile, und die Auswahl des Lasttesting-Frameworks hat großen Einfluss auf die Funktionen, die Benutzerfreundlichkeit und die Integrationen, die für den Testprozess zur Verfügung stehen. Durch die sorgfältige Abstimmung dieser Elemente können Entwicklungsteams den Lasttestansatz an die einzigartigen Eigenschaften des Spiels anpassen, die wertvollsten Erkenntnisse zur Leistung gewinnen und ihren Spielern ein reibungsloses Spielerlebnis bieten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Belastungstests in verschiedenen Entwicklungsphasen

Durch die Durchführung von explorativen Lasttests zu Beginn der Entwicklungsphasen kann die zugrunde liegende Systemarchitektur validiert werden. Dies hilft Entwicklern, fundierte Entscheidungen über die Infrastruktur, das Datenbankdesign und die Netzwerktopologie des Spiels zu treffen, bevor umfangreiche Implementierungsarbeiten durchgeführt werden. Belastungstests identifizieren Risiken und schaffen eine Leistungsbasis, wodurch der Bedarf an kostspieligen Nacharbeiten und technischen Schulden zu einem späteren Zeitpunkt im Entwicklungszyklus potenziell minimiert wird. Sie können auch ein gemeinsames Verständnis der Leistungsanforderungen des Spiels innerhalb des Teams fördern, was zu einer besseren

Zusammenarbeit und Entscheidungsfindung führt. Letztlich bilden Belastungstests in den Anfangsphasen eine solide Grundlage für ein leistungsstarkes, skalierbares und robustes Spiel und tragen so dazu bei, das Spielerlebnis insgesamt zu verbessern.

Am Ende jedes Sprints oder jeder Iteration können Lasttests die Auswirkungen der neuen Funktionen, Bugfixes und anderer Änderungen, die im letzten Zyklus eingeführt wurden, auf die Leistung bewerten. Dieser gezielte Ansatz ermöglicht es Entwicklungsteams, Regressionen oder Leistungseinbußen, die durch die neuesten Updates verursacht wurden, schnell zu erkennen, sodass sie diese Probleme beheben können, bevor sie sich weiter in der Pipeline ausbreiten, und ein konsistentes Qualitäts- und Leistungsniveau aufrechterhalten.

Vor der Bereitstellung in der Produktion unterstützen robuste Lasttests die Teams dabei, zu überprüfen, ob das System in der Lage ist, die erwarteten realen Verkehrs- und Lastbedingungen zu bewältigen. Sie können Skalierbarkeitsengpässe oder Ressourcenengpässe innerhalb der Produktionsinfrastruktur aufdecken und bieten die Möglichkeit, die Leistung des Spiels zu optimieren, sodass vom ersten Tag an ein reibungsloses und reaktionsschnelles Benutzererlebnis gewährleistet ist. Die Erkenntnisse aus den Belastungstests vor der Veröffentlichung können die Risiken am Tag der Veröffentlichung minimieren und als Grundlage für die laufende Kapazitätsplanung dienen, die den Grundstein für die langfristige Nachhaltigkeit und Skalierbarkeit des Spiels legt.

Durch Auslastungstests eines Spiels, das sich bereits in der Produktion befindet, können Teams die Leistung des Spiels überwachen und Leistungsrückgänge oder -verschlechterungen erkennen, die im Laufe der Zeit auftreten können. Auf diese Weise können sie Probleme proaktiv angehen, bevor sie sich auf das Spielerlebnis und die Nutzerbindung negativ auswirken. Darüber hinaus validieren Belastungstests in der Produktion die Wirksamkeit der implementierten Maßnahmen zur Leistungsoptimierung oder zur Skalierung der Infrastruktur. Dieser Prozess bietet Spielern ein qualitativ hochwertiges, responsives und skalierbares Spielerlebnis, auch wenn sich das Spiel weiterentwickelt und reift.

Architekturen, die Lasten erzeugen

Das Design der lastgenerierenden Architektur für Spiellasttests kann verschiedene Formen annehmen, von denen jede ihre eigenen Vorteile und Überlegungen mit sich bringt.

Auf der einfachsten Ebene können selbstverwaltete [EC2-Amazon-Instances](#) bereitgestellt und konfiguriert werden, um als Lastgeneratoren zu fungieren. Mit einem Ansatz mit Kontrollknoten und Worker-Knoten können Sie mehrere lastgenerierende Instances einrichten, von denen jede ihr eigenes Testskript ausführt und die insgesamt von einer einzigen Kontrollinstanz verwaltet werden. Die Architektur kann skaliert und mehr Last erzeugt werden, ohne die Komplexität zu erhöhen, indem

zusätzliche Worker-Knoten eingerichtet werden. Bei diesem praxisorientierten Ansatz müssen sich die Teams jedoch um die Bereitstellung, Konfiguration und Verwaltung der zugrunde liegenden Infrastruktur kümmern.

Für einen skalierbaren und orchestrierteren Ansatz können Sie [Amazon EKS](#) Kubernetes-Cluster verwenden, um den Lasttesting-Workload auf eine Flotte von containerbasierten Load Agents zu verwalten und zu verteilen. Die automatischen Skalierungsfunktionen von Kubernetes können für die Skalierung der Load generierenden Pods verwendet werden, während die Teams die zugrunde liegenden EC2 Instances in dem Cluster, der die Pods hostet, selbst konfigurieren und verwalten.

Alternativ [AWS Fargate](#) kann der serverlose Charakter von die Einrichtung von Lasttests beschleunigen und vereinfachen, indem das Infrastrukturmanagement abstrahiert wird und gleichzeitig die erforderliche Skalierbarkeit und Flexibilität gewährleistet wird. Für Hybridlösungen, bei denen bereits ein lokaler, lastgenerierender Kubernetes-Cluster vorhanden ist, aber möglicherweise zusätzliche Kapazität benötigt wird, kann [EKS Anywhere](#) beide Cluster als einen Cluster aus dem verwalten. AWS-Managementkonsole

Sie können je nach Ihren Anforderungen und Zielen auch [AWS Lambda](#) Funktionen verwenden. Lambda-Funktionen lassen sich relativ einfach einrichten und skalieren, ohne dass zusätzliche Ressourcen bereitgestellt und verwaltet werden müssen. Sie ermöglichen auch die Erstellung komplexerer und dynamischerer Testszenarien aufgrund der tiefen Integration mit anderen AWS Diensten. Lambda-Funktionen haben jedoch Beschränkungen für gleichzeitige Funktionen und Laufzeit (15 Minuten), was den Umfang und die Dauer der erreichbaren Lasttests einschränken kann. Kaltstartlatenzen können sich auch auf die Genauigkeit der Ergebnisse auswirken, und die Ressourcenbeschränkungen von Lambda sind möglicherweise nicht für sehr anspruchsvolle Lasttestworkloads geeignet.

Studios, die eine vorgefertigte Lösung verwenden möchten, können [Distributed Load Testing auf](#) verwenden. AWS Diese Lösung nutzt Amazon AWS Fargate ECS für die Bereitstellung von Containern, mit denen Simulationen von Zehntausenden verbundenen Benutzern ausgeführt werden können. Sie können dies verwenden, um Ihre Lasttestinfrastruktur schnell auf IAC-Art zu starten. AWS CloudFormation

Frameworks für Lasttests

Keine zwei Lasttest-Frameworks sind gleich aufgebaut. Einige verfügen über intuitive grafische Benutzeroberflächen für die Testerstellung, während andere vollständig auf der Befehlszeile basieren. Ein Tool kann flexibel und leistungsstark sein, erfordert jedoch Zeit und Mühe für die Konfiguration und Verwaltung, und ein anderes ist möglicherweise serverlos, aber in Bezug auf die

Tests, die es erstellen und ausführen kann, eingeschränkt. Einige verfügen über große Communitys und zahlreiche Tutorials, obwohl sie sich in der Praxis noch nicht bewährt haben, was einen starken Kontrast zu anderen darstellt, die zwar in der Produktion erprobt sind, denen es jedoch an Unterstützung oder Dokumentation durch die Community mangelt. Wählen Sie das Framework, das die richtige Balance für Sie und Ihr Team bietet. Einige beliebte Optionen sind:

- [Apache JMeter](#): Beliebtes Java-basiertes Open-Source-Framework für Lasttests aufgrund seines robusten Funktionsumfangs und seiner Benutzerfreundlichkeit. Die Fähigkeit, komplexe Benutzerszenarien zu simulieren, die breite Palette unterstützter Protokolle, die umfassende Berichterstattung und die nachgewiesene Erfolgsbilanz machen es zu JMeter einer zuverlässigen Wahl für Lasttests.
- [Locust](#): Modernes, verteiltes Lasttesting-Framework, das auf einer ereignisgesteuerten Architektur basiert, wodurch es leistungsstark und gleichzeitig ressourceneffizient ist. Tests sind in Python geschrieben, was flexible Testsznarien ermöglicht, die Tausende von leistungsstarken Bibliotheken von Drittanbietern nutzen und gleichzeitig benutzerfreundlich und einfach zu lesen sind.
- [Grafana K6](#): Leistungsstarkes Lasttest-Framework, das Benutzerfreundlichkeit mit erweiterten Funktionen kombiniert. Die Unterstützung für die Generierung verteilter Lasten, das flexible Scripting und die nahtlose Integration mit Grafana zur Datenvisualisierung machen Grafana K6 zu einer attraktiven Wahl.
- [Gatling](#): Open-Source-Framework für Lasttests, das für seine Leistung und Skalierbarkeit bekannt ist. Die auf Scala basierende, domänenspezifische Sprache (DSL) ermöglicht es Entwicklern, präzise, wartungsfreundliche Lasttestskripte zu erstellen, und die robusten Berichts- und Analysefunktionen bieten detaillierte Einblicke in das zu testende System.

Implementierungsschritte

- **Belastungstestphasen:** Führen Sie Lasttests in verschiedenen Entwicklungsphasen durch (frühe Entwicklung, Sprints, Vorproduktion und nach der Bereitstellung), um die Systemleistung zu überprüfen und Probleme zu identifizieren.
- **Lastgenerierende Architekturen:** Wählen Sie geeignete lastgenerierende Architekturen (EKSEC2, Fargate oder Lambda) auf der Grundlage von Skalierbarkeitsanforderungen, Managementpräferenzen und spezifischen Testanforderungen aus.
- **Lasttesting-Frameworks:** Wählen Sie ein Lasttesting-Framework (wie JMeter Locust, Grafana K6 oder Gatling), das Benutzerfreundlichkeit, Leistung, Flexibilität und Community-Support in Einklang bringt, um den Bedürfnissen Ihres Teams gerecht zu werden.

Optimierung im Laufe der Zeit

GAMEOPS06: Wie optimierst du dein Spiel im Laufe der Zeit?

Optimiere dein Spiel im Laufe der Zeit, indem du wichtige Metriken und Telemetriedaten überwachst, um Spielertrends, Systemleistung und Verbesserungspotenziale zu identifizieren. Aktualisiere dein Spieldesign, deine Infrastruktur und deinen Ansatz für Auslastungstests kontinuierlich auf der Grundlage dieser Erkenntnisse und passe dich gleichzeitig an neue Technologien und Frameworks an, um im Zuge der Weiterentwicklung deines Spiels eine optimale Leistung und ein optimales Spielerlebnis zu bieten.

Best Practices

- [GAMEOPS06-BP01 Überwachen Sie wichtige Spielmetriken, um Spielertrends und -muster zu identifizieren, und nutzen Sie diese Informationen, um das Spiel zu verbessern](#)
- [GAMEOPS06-BP02 Aktualisiere und passe den Lasttestansatz an, wenn sich das Spiel ändert](#)

GAMEOPS06-BP01 Überwachen Sie wichtige Spielmetriken, um Spielertrends und -muster zu identifizieren, und nutzen Sie diese Informationen, um das Spiel zu verbessern

Erfassen Sie zusätzlich zur Systemnutzung des Spielclients, zur App-Nutzung, zu Ausnahmen und Absturzdaten auch Spieltelemetriedaten, die an ein Backend-System des Spiels gesendet werden. Diese Daten sollten die Spieleraktivitäten repräsentieren, sodass du nachvollziehen kannst, wie Spieler mit den verschiedenen Funktionen im Spiel interagieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Je nach Implementierung können Spielclients Telemetriedaten an vordefinierten Spielfunktionen oder an Orten in einer Spielwelt sammeln. Die Daten werden zur Verarbeitung an den Backend-Erfassungsdienst gesendet. Wenn der Backend-Dienst nicht erreichbar ist, können die Clients die Daten lokal auf dem lokalen Gerät speichern, bis der Backend-Dienst wieder verfügbar ist. Die Spieledesigner verwenden diese Telemetriedaten, um zu überprüfen, wie die Spieler das Spiel spielen und ob es Anomalien im Spiel gibt.

Beispielsweise können Spielerbewegungen und Interaktionen mit Objekten auf einer Karte aus Telemetriedaten extrahiert und als Heatmap der Aktivitäten der Spieler im Spiel über einen festgelegten Zeitraum dargestellt werden. Anhand dieser Daten können die Spieledesigner erkennen, ob ein Gleichgewicht zwischen verschiedenen Elementen im Spiel erforderlich ist, z. B. der Stärke einer Waffe, der Stärke eines Charakters im Spiel oder der Komplexität einer Karte. Die rohen Telemetriedaten werden in der Regel gespeichert und dann verarbeitet, um Analysen zu extrahieren, die von Analysten visualisiert werden können.

Die Implementierung der [AnalyticsPipelineGame-Lösung](#) unterstützt Spieleentwickler bei der Einführung einer skalierbaren serverlosen Datenpipeline zur Erfassung, Speicherung und Analyse von Telemetriedaten, die aus Spielen und Diensten generiert werden. Die Lösung unterstützt die Streaming-Erfassung von Daten, sodass Benutzer innerhalb von Minuten Einblicke in ihre Spiele und andere Anwendungen gewinnen können.

Für die Erfassung, Speicherung, Verarbeitung und Analyse von Telemetriedaten für benutzerdefinierte Spiele bietet das Unternehmen AWS außerdem eine Reihe [spezialisierter Dienste für die Verarbeitung großer](#) Datenmengen und Analysen.

Implementierungsschritte

- Erfassung von Spieltelemetriedaten: Erfassen Sie Daten zu Spieleraktivitäten, Systemnutzung, Ausnahmen und Abstürzen, um Spielerinteraktionen zu verstehen und Probleme zu identifizieren.
- Implementieren Sie die Erfassung von Telemetriedaten: Verwenden Sie vordefinierte Spielfunktionen oder Standorte, um Telemetriedaten zu sammeln und an Backend-Dienste zu senden und lokal zu speichern, falls das Backend nicht erreichbar ist.
- Verwenden Sie AWS Analyselösungen: Nutzen Sie AWS Dienste wie die Game Analytics Pipeline für skalierbare Datenaufnahme, Speicherung und Analyse sowie spezielle Dienste zur Verarbeitung und Analyse großer Datenmengen.

GAMEOPS06-BP02 Aktualisiere und passe den Lasttestansatz an, wenn sich das Spiel ändert

Die Optimierung des Lasttestansatzes ist ein kontinuierlicher Prozess, der sich parallel zum Entwicklungszyklus des Spiels weiterentwickeln sollte. Da das Spiel an Komplexität, Nutzerbasis und Funktionsumfang zunimmt, muss die Strategie für Auslastungstests angepasst werden, um sicherzustellen, dass sie die realen Bedingungen genau simuliert und umsetzbare Erkenntnisse liefert.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Berücksichtigen Sie dabei Folgendes:

Fehlende oder veraltete Testszenarien

Wenn einem Spiel während des Entwicklungsprozesses neue Funktionen hinzugefügt werden, sollten Sie neue Belastungstestszenarien erstellen und ausführen, um die Leistung und Skalierbarkeit der neuen Funktionen zu überprüfen. In ähnlicher Weise werden Features und Funktionen häufig überarbeitet, um die Leistung zu verbessern, auf das Feedback der Spieler einzugehen oder sie an neue Designziele anzupassen. Daher müssen die Testszenarien kontinuierlich aktualisiert werden, um mit den Änderungen Schritt zu halten und den Zustand des Systems wirklich zu testen und widerzuspiegeln.

Neue Frameworks für Lasttests

Entwickler müssen möglicherweise aus verschiedenen Gründen die Frameworks für Lasttests ändern:

- Das ursprüngliche Framework ist möglicherweise nicht mehr in der Lage, die Benutzerauslastung angemessen zu simulieren oder den erforderlichen Einblick in die Systemleistung zu bieten
- Neue Spielfunktionen erfordern möglicherweise die Unterstützung von Lasttests für neue Protokolle oder Integrationspunkte APIs
- Entwickler wünschen sich möglicherweise erweiterte Funktionen, wenn sie sich mit dem Auslastungstestprozess besser auskennen
- Bevorzugung von Frameworks, die besser auf das technische Fachwissen, die Programmiersprachen oder die vorhandenen Toolchains des Teams abgestimmt sind

Durch sorgfältige Evaluierung und Anpassung im Laufe der Zeit können Entwickler den Belastungstestprozess an die sich ändernden Anforderungen des Spiels anpassen und weiterhin die erforderlichen Erkenntnisse liefern, um das allgemeine Benutzererlebnis zu optimieren und zu verbessern.

Optimierung der Kosten

Die einfache und bequeme Nutzung von Managed AWS Services kann sich insbesondere in den frühen Entwicklungsphasen als äußerst vorteilhaft erweisen. Diese Dienste abstrahieren

das zugrunde liegende Infrastrukturmanagement, sodass Teams ihre Lösung schnell einrichten und sich ausschließlich auf die Erstellung von Lasttestszenarien und die Analyse der Ergebnisse konzentrieren können. Die Nutzung von Managed Services kann jedoch häufig mit höheren Kosten verbunden sein, da sie zusätzlichen Wert und Komfort bieten, z. B. die Bereitstellung, Konfiguration und Wartung der Infrastruktur sowie die Bereitstellung von Hochverfügbarkeits-, Skalierungs- und Überwachungsfunktionen.

Wenn Teams immer reifer werden und sich mit ihrem Lasttestprozess immer wohler und sicherer fühlen, kann es vorkommen, dass die Selbstverwaltung der Infrastruktur zu zusätzlichen Optimierungen und Kosteneinsparungen führen kann. Dieser praktische Ansatz erhöht zwar den betrieblichen Aufwand, aber die direkte Kontrolle über die Rechenressourcen, Konfigurationen, Skalierungsverhalten und Ressourcennutzung kann neue Möglichkeiten zur Feinabstimmung und Kostensenkung eröffnen. Beispielsweise kann es für Teams sinnvoll sein, ihre Lasttests mit einer AWS Fargate serverlosen Architektur zu beginnen und dann später zur Selbstverwaltung der zugrunde liegenden Knoten in einem Amazon EKS-Cluster überzugehen.

Implementierungsschritte

- **Testszenarien aktualisieren:** Kontinuierlich erstellen und aktualisieren Sie Belastungstestszenarien, um neue Funktionen und überarbeitete Funktionen zu validieren und sicherzustellen, dass sie den aktuellen Stand des Spiels widerspiegeln.
- **Evaluieren Sie Lasttesting-Frameworks:** Passen Sie sich nach Bedarf an neue Frameworks an, um die Benutzerlast zu simulieren, neue Protokolle zu unterstützen und sich an das Fachwissen und die Toolchains des Teams anzupassen.
- **Kosten optimieren:** Beginnen Sie aus Gründen der Benutzerfreundlichkeit mit verwalteten AWS Diensten und ziehen Sie dann die selbstverwaltende Infrastruktur in Betracht, um Kosten zu sparen, wenn das Team mit dem Lasttestprozess immer vertrauter wird.

Ressourcen

In den folgenden Ressourcen erfahren Sie mehr über unsere Best Practices im Bereich Operational Excellence.

Dokumentation und Blogs

- [Bewährte Architekturpraktiken für Game Tech](#)
- [Verwalte dein Game Studio auf Teil AWS 1](#)

- [Verwalte dein Game Studio auf AWS pt. 2](#)
- [Verwalte dein Game Studio auf AWS pt. 3](#)
- [Einrichtung Ihrer AWS Best-Practice-Umgebung](#)
- [Multi-Account-Strategie für deine Control Tower Tower-Landezone](#)
- [Game Analytics Pipeline](#)
- [Maximieren Sie Ihre Erkenntnisse aus Spieldaten mit Game Analytics Pipeline](#)
- [Nutzung von AWS Glue Amazon Redshift Spectrum für Spielereinsichten](#)
- [Wie richte ich eine Pipeline ein CI/CD auf](#)
- [Wie Good Job Games mit AWS Build Pipeline um 43% schneller wird](#)
- [Implementierung einer Build-Pipeline für mobile Unity-Apps](#)
- [Andere relevante CI/CD Blogs](#)
- [Spiel leicht DevOps gemacht mit dem Game-Server CD Pipeline-Blog](#)
- [Harmony Games stellt mithilfe von \(\) ein vollständig benutzerdefiniertes Spiele-Backend bereit AWS Cloud Development Kit \(AWS CDK\)AWS CDK](#)
- [GameLiftBereite dich auf den Start vor](#)
- [Hybrides Gameserver-Hosting mit Amazon Gamelift Anywhere](#)
- [Beschleunigen Sie die Entwicklung von Spieleservern mit Amazon Gamelift Anywhere und dem Amazon Gamelift Agent](#)
- [So hosten Sie Ihr Unreal Engine-Spiel für unter 1\\$ pro Spieler mit Amazon Gamelift](#)
- [Neue Lösungsleitfaden für den Aufbau skalierbarer plattformübergreifender Spiele-Backends auf AWS](#)
- [Belastungstests der Pragma-Backend-Game-Engine für 1 Million gleichzeitige Benutzer auf AWS](#)
- [Wie Code Wizards Nakama von Heroic Lab an zwei Millionen gleichzeitigen Spielern getestet hat AWS](#)
- [Bewährte Methoden mit Organisationseinheiten](#)
- [AWS X-Ray](#)
- [AWS Countdown](#)
- [AWS für Games Solutions Hub](#)

Partnerlösungen

- [New Relic](#)

- [Splunk APM](#)
- [Backtrace.io](#)
- [Wachposten](#)
- [Datadog APM](#)
- [Honeycomb.io](#)

Whitepaper

- [Organisieren Sie Ihre Umgebung mithilfe mehrerer Konten](#)
- [Einführung in skalierbare Spieleentwicklungsmuster auf AWS](#)

Videos

- [YouTubeSerie: Building Games on AWS](#)
- [AWS für Spiele: Boss LEVEL Podcast](#)
- [Re:Invent 2023: Skalierung AWS für die ersten 10 Millionen Nutzer](#)
- [Re:Invent 2022: Wie Riot Games täglich 20 TB an Analysen verarbeitet AWS](#)
- [Re:Invent 2022: Wie AWS und Riot Games eine Engine für Governance-Berichte entwickelt haben](#)
- [Re:Invent 2023: Implementierung verteilter Entwurfsmuster auf AWS](#)
- [Re:Invent 2023: Mit Mortal Kombat 1 ein Multiplayer-Spiel auf Millionen skalieren](#)
- [Re:Invent 2022: Die Entwicklung der Chaos-Technik bei Netflix](#)
- [Re:Invent 2023: Bewährte Methoden für Cloud-Governance](#)
- [Re:Invent 2023: Bewährte Methoden für die Erstellung regionsübergreifender Architekturen auf AWS](#)

Schulungsmaterialien

- [Lehrplan — Erste Schritte mit AWS for Games Teil 1](#)

Sicherheit

Zur Sicherheitssäule gehört die Fähigkeit, Informationen, Systeme und Vermögenswerte zu schützen und gleichzeitig durch Risikobeurteilungen und Risikominderung einen Mehrwert für das Unternehmen zu schaffen. Aufgrund der weltweiten Sichtbarkeit und der großen Anzahl von Spielern sind Spiele ein begehrtes Ziel für Ausbeuter, Hacker und andere, die nach Möglichkeiten suchen, Systeme auszunutzen und zu missbrauchen. Dies kann oft zu einem enttäuschenden Spielerlebnis und erhöhten Kosten für den Spieleentwickler führen, wenn keine solide Sicherheitsgrundlage vorhanden ist.

Wie im [Modell der gemeinsamen Verantwortung](#) beschrieben, ist es wichtig zu verstehen, für welche Sicherheitsaspekte AWS und für welche Aspekte der Kunde verantwortlich ist, damit Sie darauf vorbereitet sind, ein solides Sicherheitsniveau aufrechtzuerhalten. Diese Säule enthält Empfehlungen zu bewährten Methoden zur Cloud-Sicherheit, die Sie bei der Entwicklung und dem Betrieb von Spielen in der Cloud berücksichtigen sollten.

Bevor Sie ein System entwerfen, müssen Sie eine Reihe von bewährten Sicherheitsmethoden festlegen, zu denen auch Zugriffskontrollen gehören. Darüber hinaus sollten Sie in der Lage sein, Sicherheitsvorfälle zu identifizieren und Ihre Systeme und Dienste zu schützen und gleichzeitig die Vertraulichkeit und Integrität der Daten durch Datenschutz zu wahren. Richten Sie ein gut definiertes und geübtes Verfahren ein, das es Ihnen ermöglicht, auf Sicherheitsvorfälle zu reagieren. Diese Tools und Techniken sind wichtig, da sie Geschäftsziele wie die Vermeidung finanzieller Verluste oder die Einhaltung regulatorischer Verpflichtungen unterstützen.

Kundenbeispiel

AnyCompany Games ist ein fiktives Spielestudio, das gerade dabei ist, seine Sicherheitslage zu verbessern. Sicherheit kann leicht zu verstehen sein, wenn es eine Erklärung für ihre direkte Anwendung gibt. AnyCompany In diesem Abschnitt werden Spiele verwendet, um die in der Säule beschriebenen bewährten Sicherheitsmethoden zu kontextualisieren

Schwerpunktbereiche

- [Designprinzipien](#)
- [Sicherheitsgrundlagen](#)
- [Kontinuierliche Sicherheit](#)
- [Identity and Access Management](#)

- [Zugriffskontrolle](#)
- [Erkennung](#)
- [Schutz der Infrastruktur](#)
- [Vorfallreaktion](#)
- [Anwendungssicherheit](#)
- [Automatisieren Sie die Sicherheit](#)
- [Modellierung von Bedrohungen](#)
- [Ressourcen](#)

Designprinzipien

Zusätzlich zu den Designprinzipien aus der Sicherheitssäule des Well-Architected Framework-Whitepapers kann das folgende Designprinzip die Sicherheit Ihrer Spiele-Workloads in der Cloud erhöhen:

- Überwachen und moderieren Sie das Nutzungsverhalten von Spielern: Erfassen und analysieren Sie Nutzungsdaten, um zu verstehen, wie Spieler mit Ihrem Spiel und Ihren sozialen Funktionen interagieren. Durch die Analyse dieser Daten können Sie missbräuchliches und unangemessenes Verhalten, das das Spielerlebnis beeinträchtigen kann, erkennen und darauf reagieren.

Sicherheitsgrundlagen

GAMESEC01: Wie implementiert man Sicherheitsgrundlagen für die Spieleentwicklung?

Spielestudios benötigen einen einzigartigen Sicherheitsansatz, der sowohl Entwicklungsumgebungen als auch Live-Spielerdienste schützt. Eine robuste AWS Sicherheitsstrategie für Spielestudios erfordert drei miteinander verbundene Komponenten: eine Struktur mit mehreren Konten, eine starke Authentifizierung und eine klare Autorisierungsstrategie mithilfe von IAM-Richtlinien. Eine AWS Struktur mit mehreren Konten ermöglicht es Studios, verschiedene Spielprojekte, Entwicklungsphasen und Toolumgebungen voneinander zu trennen. Dies gibt Studios eine detailliertere Kontrolle über Dinge wie den Zugriff auf bestimmte Umgebungen oder Dienste. Durch die Aktivierung einer starken Authentifizierung wird sichergestellt, dass Teammitglieder sicher auf Entwicklungsressourcen zugreifen können, unabhängig davon, ob sie im Studio oder remote

arbeiten, und gleichzeitig strenge Kontrollen über Quellcode, Spiele-Builds und proprietäre Tools beibehalten. Studios sollten außerdem über eine klare Autorisierungsstrategie für die Erteilung von Berechtigungen verfügen, die das Prinzip der geringsten Rechte bei IAM-Berechtigungen und -Rollen anwenden. Verwenden Sie IAM-Rollen, um den Rollen des Entwicklungsteams Berechtigungen zuzuweisen. So können Sie beispielsweise Entwicklerteams Zugriff auf AWS Dienste auf niedriger Ebene gewähren und Künstler und Designer auf bestimmte Asset-Management- und Build-Systeme beschränken. Dieser spezielle Ansatz stellt sicher, dass Spielestudios ihr geistiges Eigentum schützen, effiziente Entwicklungsabläufe aufrechterhalten und ihre Teams sicher skalieren können. Gleichzeitig erhalten Entwickler den entsprechenden Zugriff, um schnell an ihren Projekten zu arbeiten.

Best Practices

- [GAMESEC01-BP01 Verwenden Sie Rollen und Verbundzugriff anstelle des Konto-Root-Benutzers, um Aktionen in Ihrer Umgebung durchzuführen AWS](#)
- [GAMESEC01-BP02 Wird verwendet AWS Control Tower , um schnell eine Umgebung mit mehreren Konten einzurichten auf AWS](#)
- [GAMESEC01-BP03 Verwenden Sie Rollenrichtlinien mit den geringsten Rechten, die auf bestimmte Jobfunktionen zugeschnitten sind](#)
- [GAMESEC01-BP04 Verwenden Sie Rollen und Verbundzugriffsrichtlinien zusammen mit Zugriffsrichtlinien auf Kontoebene, um Zugriff auf Ihre Ressourcen zu gewähren AWS](#)
- [GAMESEC01-BP05 Verwenden Sie einen zentralen Identitätsanbieter](#)

GAMESEC01-BP01 Verwenden Sie Rollen und Verbundzugriff anstelle des Konto-Root-Benutzers, um Aktionen in Ihrer Umgebung durchzuführen AWS

Wenn Sie ein neues AWS-Konto erstellen, beginnen Sie mit einer Identität, dem so genannten Root-Benutzer, auf den Sie über die E-Mail-Adresse und das Passwort zugreifen, die dem Konto zugeordnet sind. Der Root-Benutzer hat vollständigen Zugriff auf AWS Dienste und Ressourcen innerhalb dieses Kontos. In den meisten Fällen sollten Sie es vermeiden, den Root-Benutzer für day-to-day Aufgaben zu verwenden. Wenn ein Zugriff auf Root-Ebene erforderlich ist, vergewissern Sie sich, dass dies unbedingt erforderlich ist, und stellen Sie sicher, dass zusätzliche Protokollierungs- und Schutzmaßnahmen vorhanden sind, um die Nutzung nachzuverfolgen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

In einer AWS Organizations Konfiguration hat jedes Konto immer noch seinen eigenen Root-Benutzer, aber der day-to-day Zugriff sollte stattdessen über IAM-Rollen und IAM Identity Center-Benutzer verwaltet werden. Richten Sie einen rollenbasierten Zugriff ein, der auf die Lebenszyklusphasen und Teams Ihres Spiels zugeschnitten ist. Beispielsweise benötigt das Live-Operations-Team möglicherweise Berechtigungen, um Ereignisse im Spiel zu verwalten, während Entwickler Zugriff auf Push-Updates benötigen. Wenn Sie mit Diensten oder Partnern von Drittanbietern zusammenarbeiten, sollten Sie Verbundzugriff verwenden, um eine sichere Zusammenarbeit zu ermöglichen, ohne sensible Infrastrukturen preiszugeben. Dieser Ansatz stellt sicher, dass jeder Benutzer oder Partner nur den Zugriff hat, den er benötigt, und gleichzeitig die Sicherheit der Infrastruktur und der Spielerdaten Ihres Spiels gewährleistet.

Kundenbeispiel

AnyCompany Spiele haben bei der Entwicklung ihres neuen Spiels eine rollenbasierte Zugriffskontrolle implementiert. Durch die Verwendung spezifischer IAM-Rollen für ihre unterschiedlichen Entwicklungsteams vermeiden sie die Verwendung gemeinsamer Anmeldeinformationen. Dieses Setup ermöglicht es einem Entwicklerteam, eine Rolle für zentrale Spielsysteme zu übernehmen, während die Rolle des Content-Teams nur Zugriff auf Asset-Management-Services hat.

Implementierungsschritte

- Verwenden Sie den Root-Benutzer nicht, nachdem Sie ein Konto eingerichtet haben, es sei denn, dies ist unbedingt erforderlich. Erstellen Sie das Konto, sichern Sie den Root-Benutzer, erstellen Sie sofort die erforderlichen IAM-Administrationsrollen und weisen Sie diese Rolle einem Verbundbenutzer zu.
- Verwenden Sie den Root-Benutzer nur, wenn Sie [eine begrenzte Anzahl von Aufgaben ausführen müssen, die nur dem Root-Benutzer zur Verfügung stehen](#). Zu diesen Aufgaben gehören beispielsweise das Ändern der E-Mail-Adresse Ihres Root-Benutzers und das Ändern Ihres AWS Supportplans.

GAMESEC01-BP02 Wird verwendet AWS Control Tower , um schnell eine Umgebung mit mehreren Konten einzurichten auf AWS

Wenn Sie die Nutzung AWS mit nur einem einzigen Konto beginnen, werden Sie feststellen, dass Ihr Spielestudio mit fortschreitender Spieleentwicklung immer weiter wächst. Mit einem einzigen

AWS-Konto Konto stoßen Sie möglicherweise allmählich an die Leistungsgrenzen, oder Ihre Kosten für verschiedene Projekte und Workloads werden möglicherweise komplexer. Die Erstellung verschiedener Konten für verschiedene Spieletitel und Umgebungen ermöglicht es Teams, mit neuen Funktionen zu experimentieren, Servicebeschränkungen zu umgehen und den Sicherheitsstatus und die Einhaltung von Vorschriften aufrechtzuerhalten. Durch die Implementierung einer Strategie für mehrere Konten können Sie von der Verteilung der Servicelimits auf mehrere Konten profitieren und Einblicke in Ihre AWS Kosten gewinnen. AWS

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Es ist ein weit verbreitetes Missverständnis, dass die Verwendung mehrerer AWS-Konten Benutzer automatisch verwirrender und zeitaufwändiger ist. Vielmehr kann die Nutzung von AWS Diensten, die die Verwaltung mehrerer Konten erleichtern sollen, Ihrem Spielestudio helfen, weniger Zeit mit der Verwaltung Ihrer Konten zu verbringen.

Sie können diesen Dienst verwenden AWS Control Tower , um eine AWS Umgebung mit mehreren Konten sicher bereitzustellen. Control Tower wird empfohlen, wenn Sie eine neue AWS Umgebung aufbauen, Ihre Reise beginnen oder ganz neu darin sind AWS. AWS Während des kurzen Einrichtungsprozesses können Sie andere AWS Dienste integrieren, die mit der Verwaltung von Konten und Benutzerzugriffen befasst sind, z. B. AWS Organizations Service Catalog und AWS IAM Identity Center.

Kundenbeispiel

AnyCompany Die Spiele wurden zunächst von einem einzigen System AWS-Konto aus betrieben und stießen bei einem wichtigen Betatest auf mehrere Hindernisse, als ein Mitglied des Entwicklungsteams ihrer Spiele die EC2 Leistungsgrenzen erreichte. Gleichzeitig hatte ihr Entwicklungsteam für ein anderes Spiel Probleme mit der Ressourcenzuweisung für ihre automatisierte Test-Pipeline. Die Situation erreichte einen kritischen Punkt, als AnyCompany Games die Kosten nicht genau zwischen den Projekten trennen konnte, was es schwierig machte, für die Entwicklung jedes Spiels ein Budget festzulegen.

AnyCompany Games implementierte daraufhin eine Strategie für mehrere Konten mithilfe von AWS Control Tower. Sie erstellten separate Konten für jedes Spieleprojekt mit unterschiedlichen Entwicklungs-, QA- und Produktionsumgebungen. Durch diese Trennung auf Kontoebene werden die Daten und Ressourcen jedes Projekts isoliert, sodass Teams, die an einem Spiel arbeiten, nicht auf Ressourcen eines anderen zugreifen oder diese ändern können. Dadurch wurde eine zentralisierte

Abrechnungsstruktur eingeführt AWS Organizations, in der die Infrastrukturkosten der einzelnen Spiele klar ausgewiesen wurden, und es wurden unternehmensweite Zugriffsrichtlinien eingeführt.

Implementierungsschritte

- Verwenden Sie AWS Control Tower, um eine automatisierte Umgebung mit mehreren Konten einzurichten.
- Organisieren Sie Konten auf der Grundlage von Umgebungen (wie Entwicklung, Qualitätssicherung und Produktion).
- Verwenden Sie AWS IAM Identity Center und Service Catalog, um Benutzerberechtigungen zu zentralisieren und die Ressourcenbereitstellung für alle Konten zu optimieren.

GAMESEC01-BP03 Verwenden Sie Rollenrichtlinien mit den geringsten Rechten, die auf bestimmte Jobfunktionen zugeschnitten sind

Die Konfiguration von IAM-Richtlinien ist ein wesentlicher Bestandteil der Schaffung einer soliden Sicherheitsgrundlage. Gewähren Sie die Berechtigungen mit IAM Richtlinien nur die zum Ausführen einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. QA-Teams benötigen beispielsweise Zugriff auf Änderungen in den Testumgebungen, sollten aber nicht in der Lage sein, die Produktionsumgebung zu ändern.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Sie könnten mit umfassenden Berechtigungen beginnen, wie z. B. [verwalteten Richtlinien](#), während Sie sich mit den Berechtigungen befassen, die für Ihre Arbeitslast oder Ihren Anwendungsfall erforderlich sind. Mit zunehmender Reife Ihres Anwendungsfalls können Sie daran arbeiten, die Berechtigungen zu reduzieren, die Sie gewähren, um auf die geringsten Berechtigungen hinzuarbeiten.

Implementierungsschritte

- Halten Sie sich bei der Erstellung von IAM-Rollen für Benutzer und Anwendungen an die Praxis der Berechtigungen mit den geringsten Rechten.

- Verwenden Sie AWS verwaltete Richtlinien, um schnell umfassenden Zugriff zu gewähren und gleichzeitig die spezifischen Berechtigungen zu ermitteln, die Teams oder Anwendungen zur Ausführung ihrer Aufgaben benötigen.
- Studios können auch die [Richtliniengenerierung von IAM Access Analyzer](#) verwenden, um benutzerdefinierte IAM-Richtlinien auf der Grundlage von CloudTrail Ereignissen zu generieren, die Aktionen und Dienste identifizieren, die von einem IAM-Eintrag verwendet werden.
- Überprüfen Sie regelmäßig die IAM-Richtlinien und bearbeiten Sie zu freizügige Richtlinien.

GAMESEC01-BP04 Verwenden Sie Rollen und Verbundzugriffsrichtlinien zusammen mit Zugriffsrichtlinien auf Kontoebene, um Zugriff auf Ihre Ressourcen zu gewähren AWS

Neue AWS Benutzer verwenden IAM-Richtlinien häufig nur, wenn sie anderen Zugriff gewähren. Wenn Sie jedoch verwenden, sollten Sie überlegen AWS Organizations, wie Sie Richtlinien zur Servicesteuerung zusammen mit IAM-Richtlinien verwenden können, um Ihren Studio-Teammitgliedern und Auftragnehmern die erforderlichen Zugriffsebenen zu gewähren.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Sie können IAM-Richtlinien erstellen, um den Zugriff auf AWS Dienste oder API-Aktionen, die mit funktionieren, zuzulassen oder zu verweigern. AWS Identity and Access Management Sie können nur auf IAM-Identitäten wie Benutzer, Gruppen oder Rollen angewendet werden. Beispielsweise kann eine IAM-Richtlinie verwendet werden, um einem Benutzer nur Lesezugriff auf Amazon S3 zu gewähren.

Richtlinien zur Servicekontrolle (SCPs) sind Leitplanken für Sie. AWS-Konten Ein SCP gewährt keine Berechtigungen, sondern sie werden verwendet, um Aktionen im Zusammenhang mit AWS Diensten für einzelne Mitgliedskonten einzuschränken. Beispielsweise kann ein SCP einem den Zugriff auf eine AWS-Konto bestimmte Region verweigern.

Wenn eine Maßnahme ergriffen wird, wird die entsprechende IAM-Richtlinie in Kombination mit der bewertet. SCPs In Anlehnung an das vorherige Beispiel, wenn eine Rolle versucht, eine EC2 Instance auszuführen, gibt IAM an, ob sie erlaubt sind („Allow“ für ec2:RunInstances) und sie bestimmt, ob ihre Wahl der Region gültig ist („us-east-1“ ist erlaubt, aber „us-west-1“ wird vom SCP verweigert). SCPs

Durch die Kombination von IAM-Richtlinien SCPs können Sie sicherstellen, dass jeder, der auf Ihre AWS Ressourcen zugreift, nur die entsprechenden Berechtigungen erhält, die er benötigt. Dies ist besonders wichtig, wenn sich deine Ressourcen AWS-Konten und deine Ressourcen über mehrere Regionen erstrecken, aber nicht jeder in deinem Game-Studio auf alle zugreifen muss.

Sie können die IAM-Richtlinien so anpassen, dass sie bestimmten Teams spezifische Berechtigungen für die Aktualisierung von Spielkonfigurationen, die Verwaltung von Spielerdaten, die Konfiguration von Werbeveranstaltungen und die Moderation von nutzergenerierten Inhalten gewähren. In der Zwischenzeit können Sie damit unternehmensweite Kontrollen durchsetzen SCPs , die für den Spielbetrieb von entscheidender Bedeutung sind. Dazu gehören unter anderem die Beschränkung der Bereitstellung auf zugelassene Regionen, in denen das Spiel betrieben wird, um unbefugten Zugriff auf sensible Spielerdatenspeicher zu verhindern, Compliance-Anforderungen durchzusetzen und die Kosten zu kontrollieren, indem die Nutzung der Dienste auf alle Entwicklerkonten beschränkt wird.

Implementierungsschritte

- Verwenden Sie IAM-Richtlinien, um die Berechtigungen für einzelne Benutzer, Gruppen oder Rollen zu verwalten.
- Verwenden Sie Dienststeuerungsrichtlinien (SCPs) AWS Organizations , um Berechtigungen auf Kontoebene durchzusetzen.
- Kombinieren Sie IAM-Richtlinien SCPs , um nur bestimmten Benutzern und Konten den erforderlichen Zugriff zu gewähren.

Ressourcen

- [Richtlinien und Berechtigungen in IAM AWS](#)
- [Service-Kontrollrichtlinien](#)
- [Von AWS verwaltete Richtlinien für Auftragsfunktionen](#)

GAMESEC01-BP05 Verwenden Sie einen zentralen Identitätsanbieter

Ein zentraler Identitätsanbieter dient als zentrale Quelle für die Speicherung und Verwaltung von Benutzeranmeldedaten, Identitäten, Berechtigungen und Authentifizierung.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Verwenden Sie einen zentralen Identitätsanbieter, um Ihren Benutzerauthentifizierungsprozess zu optimieren, konsistente Sicherheitsrichtlinien durchzusetzen und Ihre Benutzerverwaltung für alle Anwendungen zu vereinfachen. AWS-Konten Durch einen zentralisierten Ansatz müssen Benutzeridentitäten und Anmeldeinformationen nicht mehr getrennt verwaltet werden, wodurch das Risiko von Inkonsistenzen, Redundanzen und anderen Sicherheitslücken verringert wird. Die Konsolidierung von Benutzeridentitäten und Authentifizierung an einem zentralen Ort ermöglicht auch eine bessere Transparenz, Kontrolle und Überprüfbarkeit für Ihre gesamte Umgebung. AWS

Kundenbeispiel

AnyCompany Spiele standen bei der Verwaltung des Entwicklerzugriffs in ihrer schnell wachsenden Infrastruktur vor erheblichen Herausforderungen. AWS Ihr Entwicklungsteam wuchs von 50 auf 200 Mitarbeiter in drei großen Titeln. Anfänglich verwaltete jedes Projektteam seine eigenen AWS Zugangsdaten, was zu inkonsistenten Sicherheitspraktiken, verzögertem Onboarding neuer Entwickler und gelegentlichen Sicherheitsvorfällen führte.

Das Studio implementierte AWS IAM Identity Center als zentralen Identitätsanbieter und konsolidierte so die Benutzerverwaltung in einem einzigen System. Sie verknüpften es mit ihrem bestehenden Unternehmensverzeichnis, sodass Entwickler dieselben Unternehmensanmeldedaten für AWS den Zugriff verwenden konnten. Jetzt verwenden Entwickler ihr einziges, vorhandenes Firmen-Login, um den AWS Zugriff zu erhalten, den sie für die Erledigung ihrer Arbeit benötigen

Implementierungsschritte

- Erwägen Sie, AWS IAM Identity Center als zentralen Identitätsanbieter zu verwenden. Dies ermöglicht eine konsistente Zugriffsverwaltung für alle Ihre AWS-Konten Anwendungen, bietet Ihren Mitarbeitern eine Single-Sign-On-Authentifizierung und vereinfacht die Prüfung des Benutzerzugriffs auf Ihre AWS Anwendungen. IAM Identity Center stellt auch eine Verbindung zu bestehenden Unternehmensidentitäten von unterstützten Identitätsanbietern her.

Kontinuierliche Sicherheit

GAMESEC02: Wie erreichen, pflegen und überwachen Sie fortlaufende bewährte Sicherheitsverfahren?

Die Einhaltung der besten Sicherheitspraktiken ist für Unternehmen aller Branchen, insbesondere aber in der Spielebranche, von größter Bedeutung. Die Spielebranche ist darauf angewiesen, das Vertrauen der Spieler zu pflegen und aufrechtzuerhalten und sich einen guten Ruf aufzubauen, und selbst geringfügige Sicherheitsprobleme können dieses Vertrauen schnell untergraben.

Darüber hinaus erfordert der globale Charakter der Spielebranche die Einhaltung verschiedener Branchenvorschriften und -standards in Bezug auf Datenschutz, Verbraucherschutz und Sicherheit in den Regionen, in denen Spiele angeboten werden. Faires und sicheres Gameplay ist ein weiterer wichtiger Aspekt, der die Bedeutung robuster Sicherheitsmaßnahmen unterstreicht. Cheaten, Hacken und andere Formen der Ausnutzung von Spielen können das Spielerlebnis für legitime Spieler beeinträchtigen. Deshalb sind strenge Sicherheitskontrollen unerlässlich, um die Integrität des Spiels zu wahren und gleiche Wettbewerbsbedingungen für die Teilnehmer zu fördern.

Best Practices

- [GAMESEC02-BP01 Verwenden Sie einsatzbereite Vorlagen für Standardsicherheitspraktiken](#)
- [GAMESEC02-BP02 Verwenden Sie automatisierte Abhilfemaßnahmen, wenn ein Sicherheitsereignis eintritt](#)

GAMESEC02-BP01 Verwenden Sie einsatzbereite Vorlagen für Standardsicherheitspraktiken

Ready-to-deploy-Vorlagen bieten eine proaktive und agile Methode zur Bewertung Ihres Sicherheitsstatus in der Cloud. Vorkonfigurierte Vorlagen bewerten Ihre Cloud-Sicherheit und implementieren notwendige Änderungen umgehend. Die Vorlagen umfassen eine Vielzahl von Best Practices für verschiedene Technologien und allgemein anerkannte Sicherheits-Frameworks. Die Verwendung von Vorlagen kann Spielestudios dabei helfen, konsistente Infrastrukturkonfigurationen aufrechtzuerhalten, insbesondere da sie skalieren und zusätzliche hinzufügen können, AWS-Konten um neue Workloads zu unterstützen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Durch die Nutzung von AWS Diensten und die Implementierung von ready-to-deploy Vorlagen können Spieleentwickler ihre Cloud-Sicherheitslage proaktiv bewerten und verbessern, ihr geistiges Eigentum schützen, Spielerdaten schützen und eine sichere Spielelandschaft durch regelmäßige Sicherheitsbewertungen und kontinuierliche Überwachung fördern, um potenzielle Sicherheitslücken umgehend zu erkennen und zu beheben.

Kundenbeispiel

AnyCompany Spiele standen vor einer großen Herausforderung, als sie sich darauf vorbereiteten, ihren nächsten Titel in der europäischen Industrie auf den Markt zu bringen. Sie stellten fest, dass ihre bestehenden Datenverarbeitungspraktiken nicht den DSGVO-Anforderungen entsprachen. Sie wandten sich an AWS Security Hub CSPM und AWS Config und ihre ready-to-deploy Vorlagen, um eine Lösung zu finden. Das Team implementierte das DSGVO-spezifische Konformitätspaket AWS Config, mit dem die bestehende Infrastruktur automatisch anhand der DSGVO-Standards bewertet wurde. Bei diesem ersten Scan wurden mehrere kritische Lücken aufgedeckt, wie z. B. unsachgemäße Richtlinien zur Datenspeicherung und unzureichende Zugriffskontrollen darüber, wo Spielerdaten gespeichert wurden. Mithilfe der vordefinierten Regeln der Vorlage hat AnyCompany Games die erforderlichen Änderungen schnell umgesetzt. Darüber hinaus ermöglichten die laufenden automatisierten Konformitätsprüfungen, die durch die Vorlage bereitgestellt wurden, dem kleinen Team, die DSGVO-Konformität mühelos aufrechtzuerhalten, auch wenn das Spiel ständig aktualisiert und erweitert wurde.

Implementierungsschritte

- Verwenden Sie Vorlagen für Standardsicherheitspraktiken, z. B. für verwaltete Regeln und Konformitätspakete AWS Config sowie für Standards. AWS Security Hub CSPM
- Lesen Sie die Einzelheiten der [Security Hub CSPM-Standards](#), um festzustellen, welche den Sicherheitsanforderungen Ihres Spielestudios am besten entsprechen.

GAMESEC02-BP02 Verwenden Sie automatisierte Abhilfemaßnahmen, wenn ein Sicherheitsereignis eintritt

Mithilfe automatisierter Problembehebungstechniken können Spieleentwickler ihre Spieleinfrastruktur proaktiv schützen und warten und die potenziellen Auswirkungen eines Sicherheitsvorfalls minimieren. Wenn ein Sicherheitsproblem erkannt wird, verwenden Sie ein Runbook, das Ihnen als Leitfaden für die Reaktion auf die Situation dient. Automatisieren Sie diese Reaktionen nach Möglichkeit, um Probleme schneller zu beheben und ihre Auswirkungen zu verringern. Dies verbessert das Spielerlebnis, indem die Wahrscheinlichkeit von Ausfallzeiten und Spielunterbrechungen verringert wird.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Wenn Sie sich darauf vorbereiten, auf Sicherheitsprobleme zu reagieren, wird nicht nur das Spielerlebnis gewährleistet, sondern auch die Einhaltung der verschiedenen Compliance- und Regulierungsstandards gewährleistet. Darüber hinaus skaliert der Einsatz automatisierter Sicherheitsreaktionen Ihre Sicherheitsabläufe, wenn Ihre Workloads zunehmen. AWS bietet Services zur Identifizierung und Automatisierung der Reaktion auf diese Vorfälle.

Kundenbeispiel

AnyCompany Bei Spielen kam es zu einem kritischen Sicherheitsvorfall, als während eines routinemäßigen Updates der Asset-Pipeline versehentlich ein S3-Bucket mit unveröffentlichten Charaktermodellen und Texturen für das kommende Spiel veröffentlicht wurde. Das automatisierte Sicherheitssystem erkannte die Änderung der Bucket-Berechtigungen innerhalb von Minuten nach der Änderung. Das System führte sofort sein Reparatur-Runbook aus: Der Bucket wurde wieder in den privaten Status versetzt, Zugriffsversuche während des Offenlegungsfensters protokolliert, das Sicherheitsteam benachrichtigt und ein detailliertes CloudTrail Protokoll der Berechtigungsänderungen erstellt.

Implementierungsschritte

- Verwenden Sie die AWS Lösung [Automated Security Response on](#), um Automatisierungs-Runbooks zu implementieren, in denen die Aktionen definiert werden, die automatisch als Reaktion auf Sicherheitsereignisse in ergriffen werden. AWS Security Hub CSPM

Ressourcen

- [AWS for Games Blog — Verwalte dein Game Studio auf AWS: Teil 1](#)
- [AWS für Games Blog — Dein Game Studio verwalten — AWS Teil 2](#)
- [Registrierte eine bestehende Organisationseinheit bei AWS Control Tower](#)
- [AWS-Konto Root-Benutzer](#)
- [Aufgaben, für die Root-Benutzeranmeldedaten erforderlich sind](#)
- [Automatisierte Sicherheitsreaktion auf AWS](#)

Identity and Access Management

GAMESEC03: Wie verwaltet man die Identitäts- und Zugangsverwaltung von Spielern?

Bei der Entwicklung eines Spiels müssen Sie festlegen, wie Sie Spielern Zugriff auf Ihr Spiel und die damit verbundenen Dienste gewähren. Im folgenden Abschnitt werden Überlegungen zum Design untersucht, darunter Spielerauthentifizierung, Autorisierung und Multi-Faktor-Authentifizierung.

Best Practices

- [GAMESEC03-BP01 Bestimmen Sie, wie Sie den Zugriff von Spielern auf die Umgebung und die Ressourcen Ihres Spiels identifizieren und kontrollieren möchten](#)
- [GAMESEC03-BP02 Authentifiziere Anfragen, die an den Backend-Dienst deines Spiels gesendet werden](#)
- [GAMESEC03-BP03 Verwende deinen Spiel-Backend-Service, um Spielieranfragen zur Teilnahme an einem Multiplayer-Spiel zu validieren](#)
- [GAMESEC03-BP04 Setzen Sie strenge Sicherheitsrichtlinien für Spielerbenutzerkonten durch, indem Sie ein sicheres Passwort verlangen](#)
- [GAMESEC03-BP05 Bieten Sie Spielern die Möglichkeit, die Multi-Faktor-Authentifizierung \(MFA\) für ihre Konten einzurichten](#)

GAMESEC03-BP01 Bestimmen Sie, wie Sie den Zugriff von Spielern auf die Umgebung und die Ressourcen Ihres Spiels identifizieren und kontrollieren möchten

Diese Entscheidung wird von Ihrer Strategie zur Spielerakquise und Monetarisierung, der Spielerfahrung und anderen Faktoren beeinflusst, wie z. B. den vorhandenen Funktionen, die möglicherweise von Ihren Publishing-Partnern bereitgestellt werden. Ein Spiel kann zum Beispiel Käufe erfordern und verlangen, dass ein Spieler ein Benutzerprofil erstellt, um Echtgeld-Zahlungsmethoden mit seinem Konto zu verknüpfen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Alternativ könnte ein Spiel die Eintrittsbarriere für Erstspielererfahrungen verringern, indem die Notwendigkeit entfällt, vor dem Spielen ein Benutzerkonto zu erstellen, wodurch die Wahrscheinlichkeit erhöht wird, dass ein Spieler das Spiel zum ersten Mal ausprobiert. In der Regel implementieren Spiele für ihr Spiel eine oder mehrere Kombinationen von Ansätzen zur Spieleridentität und zur Zugriffsverwaltung.

Nicht authentifizierter oder anonymer Zugriff

Diese Zugriffsebene ist in Situationen nützlich, in denen ein Spieler für ein Spiel kein neues Benutzerkonto erstellen oder sich mit seiner Identität in sozialen Netzwerken und Spielesystemen verknüpfen muss. Dies ist der einfachste und schnellste Weg für einen Spieler, mit dem Spielen eines Spiels zu beginnen, und ist besonders nützlich bei Handyspielen, bei denen ein Spieleentwickler möglicherweise die Eintrittsbarriere für das erste Spielerlebnis verringern möchte.

Wenn Sie in diesem Zugriffsszenario die Nutzung anhand der Spielinstallation identifizieren möchten, können Sie den Spielclient so programmieren, dass er eine eindeutige Kennung generiert und auf dem Gerät des Spielers speichert. Diese eindeutige Kennung wird verwendet, um den Spieler bei allen Spielsitzungen auf seinem Gerät zu identifizieren und Analyseberichte über die Nutzung im Laufe der Zeit zu ermöglichen. Wenn sich ein Spieler später dafür entscheidet, ein Konto zu erstellen, können Sie sein neues Benutzerkonto mit seiner zuvor generierten eindeutigen Kennung verknüpfen. Dadurch wird die Identität des neuen Spielers mit seiner bisherigen Nutzung verknüpft, zu der auch Statistiken und Spielerfolge gehören können.

Wenn ein Spieler nicht irgendwann ein Konto erstellt und verknüpft, kann das Gerät, das der Spieler zur Interaktion mit dem Spiel verwendet, eindeutig identifiziert werden, aber wiederherstellbare Informationen über den Spieler werden nicht gesammelt und gespeichert. Wenn also der Spieler sein Gerät kaputt macht oder verliert, gehen auch die zuvor mit dem Gerät verknüpften gespeicherten Daten verloren und können möglicherweise nicht wiederhergestellt werden.

Authentifizierung mit Benutzername und Passwort

Ein Spiel kann es Spielern ermöglichen, ihre eigenen Benutzerkonten mit einem Benutzernamen und einem Passwort zu erstellen, die im Backend des Spiels gespeichert sind. Dies kann der Fall sein, wenn ein Spieleentwickler mit einem Spieleverleger zusammenarbeitet, der bereits über ein vorhandenes Spielerkontensystem verfügt, in das sich der Entwickler integrieren kann. Alternativ möchte ein Entwickler, der seine eigenen Spiele veröffentlicht, das Spielerlebnis vereinfachen, indem er es Spielern ermöglicht, ein einziges Benutzerkonto für den Zugriff auf alle von ihnen veröffentlichten Spiele zu erstellen.

Authentifizierung und Kontoverknüpfung mit sozialen Netzwerken und Spielesystemen von Drittanbietern

Bei Online-Spielen und Spielen mit sozialen Funktionen ist es üblich, einen Verbund von Identitätsanbietern durch Drittanbieter bereitzustellen, um das Spielerlebnis zu vereinfachen. Anstatt Spieler zu bitten, für die Authentifizierung eine Kombination aus Benutzername und Passwort zu erstellen, können Sie den Identitätsverbund verwenden, um es Spielern zu ermöglichen, sich über ihre Drittanbieterkonten in sozialen Netzwerken und Spielesystemen zu authentifizieren. Dieser Anmeldevorgang vereinfacht die Anmeldung und Registrierung für Spieler. Es bietet auch eine bequeme Alternative zur obligatorischen Kontoerstellung und eine reibungslose Methode für Spieler, auf Spiele zuzugreifen.

Für Spieleentwickler kann ein förderierter Anmeldeprozess einen optimierten Ablauf bei der Spielerverifizierung bieten. Es kann auch eine zuverlässigere Methode zur Verwaltung von Spielerdaten bieten, die für die Personalisierung verwendet werden. Das liegt daran, dass du Spieler nicht bitten musst, dir bestimmte Daten zur Verfügung zu stellen, die sie wahrscheinlich bereits an den externen Identitätsanbieter weitergegeben haben. Darüber hinaus bieten diese Systeme die Integration zusätzlicher sozialer Funktionen, wie z. B. die Möglichkeit, Spieler mit ihren Freunden zu verbinden.

Implementierungsschritte

- Nutze unauthentifizierten oder anonymen Zugriff, um Hindernisse für Erstspieler zu verringern, indem du eine eindeutige Geräteerkennung generierst, um die Nutzung nachzuverfolgen und die Kontoverlinkung zu einem späteren Zeitpunkt zu ermöglichen.
- Implementieren Sie die Authentifizierung mit Benutzername und Passwort für spezielle Benutzerkonten, indem Sie bestehende Spielerkontosysteme verwenden oder ein einheitliches Spielerlebnis für alle Spiele schaffen.
- Integrieren Sie externe Identitätsanbieter für die förderierte Authentifizierung, vereinfachen Sie Anmeldeprozesse und ermöglichen Sie den Zugriff auf soziale Funktionen und Personalisierungsdaten.

GAMESEC03-BP02 Authentifiziere Anfragen, die an den Backend-Dienst deines Spiels gesendet werden

Durch die Authentifizierung von Anfragen, die an deinen Spiele-Backend-Dienst gesendet werden, kann verhindert werden, dass unerwünschte Anfragen erfolgreich sind.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Du solltest Spielern einen Authentifizierungsdienst zur Verfügung stellen, der sichere, kurzlebige Token, wie z. B. ein JSON Web Token (JWT), an den Spielclient zurückgibt, wenn sich ein Spieler erfolgreich authentifiziert hat.

Diese Token können Behauptungen enthalten, die Spielerattribute und andere relevante Metadaten enthalten. Diese relevanten Metadaten können in nachfolgenden Anfragen verwendet werden, die vom Spielclient an dein Spiel-Backend gesendet werden, um Anfragen zu authentifizieren und sie im Kontext des authentifizierten Spielers zu autorisieren.

Sie haben die Wahl, entweder Ihr eigenes Spielerauthentifizierungssystem zu entwerfen und aufzubauen, was eine kontinuierliche Verbesserung und Wartung erfordern würde, oder Sie können die skalierbaren und sicheren Funktionen für die Benutzerregistrierung, Anmeldung und Zugriffskontrolle verwenden, die von [Amazon](#) Cognito bereitgestellt werden.

Amazon Cognito Cognito-Benutzerpools enthalten ein Benutzerverzeichnis für Authentifizierung und Autorisierung. Ein Benutzerpool bietet APIs die Möglichkeit, Workflows für die Registrierung, Anmeldung und das Zurücksetzen von Passwörtern in Ihr Spiel zu integrieren. Diese können mit Identitätsanbietern von Drittanbietern integriert werden. [Application Load Balancers](#) und [Amazon API Gateway](#) bieten beide Integrationen mit Cognito, um die Benutzerauthentifizierung für Anfragen zu integrieren, die an Ihre benutzerdefinierten Spiele-Backends gesendet werden, die mit diesen Services gehostet werden.

Wenn Ihr Spiel anonymen Zugriff unterstützt und Sie einen Spieler nicht authentifizieren können, können Sie einen Client-Authentifizierungsansatz verwenden, um bei der Integration in Ihr Spiel-Backend ein sichereres Erlebnis zu bieten. Wenn dein Spielclient AWS Dienste direkt nutzt, müssen Anfragen an diese Dienste mit Anmeldeinformationen signiert werden. Um Anmeldeinformationen für Ihren Spieleclient für nicht authentifizierte Benutzer bereitzustellen, können Sie das AWS SDK verwenden, um kurzlebige Anmeldeinformationen aus [Amazon Cognito Cognito-Identitätspools](#) abzurufen, mit denen Sie Ihre Anfragen an Dienste signieren können. AWS Diese Anmeldeinformationen können von Ihrem Spielclient aus aktualisiert werden.

Neben der direkten Integration in das AWS SDK über den Spielclient können Sie auch Ihr eigenes Spiel-Backend erstellen, indem Sie einen Dienst wie [Amazon API Gateway](#) verwenden, der benutzerdefinierte Autorisierung unterstützt. Indem Sie Ihren eigenen Backend-Service für Spiele entwerfen, können Sie mit benutzerdefinierter serverseitiger Logik die maßgebliche Kontrolle über Anfragen erlangen.

Weitere Informationen zum Aufbau eines Backend-Service für Spiele, die über Amazon gehostet werden GameLift, finden Sie unter [Gestalten Sie Ihren Spieleclient-Service](#).

Kundenbeispiel

AnyCompany Spiele haben die Sicherheit ihres nächsten Titels durch die Einführung eines verwalteten Authentifizierungs- und Autorisierungsansatzes verbessert. Anstatt ein benutzerdefiniertes System mit Benutzernamen und Passwörtern beizubehalten, nutzten sie Amazon Cognito Cognito-Benutzerpools, um die Registrierung und Anmeldung von Spielern zu verwalten, und Identitätspools, um den anonymen Zugriff für Spieler zu ermöglichen, die den Trainingsmodus ausprobieren, bevor sie ein Konto erstellen. Sie haben auch eine benutzerdefinierte Autorisierungslogik innerhalb des Spiels implementiert, um in Cognito definierte Administratorrollen zu erkennen und diesen Benutzern Zugriff auf spezielle Verwaltungsfunktionen im Spiel zu gewähren.

Implementierungsschritte

- Verwenden Sie Amazon Cognito Cognito-Benutzerpools, um die Authentifizierung mit sicheren Token zu verwalten und Funktionen wie Registrierung JWTs, Anmeldung und Kennwortzurücksetzung zu aktivieren.
- Rufen Sie kurzlebige Anmeldeinformationen aus Amazon Cognito Cognito-Identitätspools ab, damit anonyme Benutzer sicher mit AWS Diensten interagieren können.
- Implementieren Sie benutzerdefinierte Spiele-Backends mithilfe von Amazon API Gateway für eine benutzerdefinierte serverseitige Authentifizierungslogik.

GAMESEC03-BP03 Verwende deinen Spiel-Backend-Service, um Spielern Anfragen zur Teilnahme an einem Multiplayer-Spiel zu validieren

In Mehrspieler-Spielen nimmt ein Spieler normalerweise an einer Spielsitzung teil, indem er eine Option direkt aus einer Liste verfügbarer Sessions auswählt, oder er reicht eine Anfrage ein, um ein Match zu finden. Bei letzterem Ansatz ist der Spieleentwickler dafür verantwortlich, eine geeignete Spielsitzung ausfindig zu machen und die Verbindungsinformationen (in der Regel eine IP-Adresse und eine Portnummer) an den Spielclient des Spielers weiterzuleiten. Die Implementierung kann je nach Spielgenre, das Sie entwickeln, variieren. Unabhängig davon ist es eine bewährte Sicherheitsmethode, die Anfrage eines Spielers, einem Spiel beizutreten, serverseitig zu überprüfen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

In einem sitzungsbasierten Multiplayer-Spiel sollte beispielsweise die Anfrage eines Spielers, an einer Spielsitzung teilzunehmen, von Ihrer Gameserver-Software mit Ihrem Matchmaking-Dienst im Spiel-Backend validiert werden, bevor die Verbindung zum Server autorisiert wird. Wenn ein Spieler die Teilnahme an einer Spielsitzung anfordert, sollte der Spieleserver die Anfrage nach einer eindeutigen Kennung überprüfen, z. B. nach einer Spielersitzungs-ID und einem vom Server generierten Ticket, die zuvor von Ihrem Spiel-Backend-Matchmaking-Service an den Spielclient gesendet wurden.

Sobald die Verbindung zum Spielserver hergestellt ist, kann Ihre serverseitige Software diese Informationen verwenden, um beim Matchmaking-Dienst zu überprüfen, ob die Verbindungsanfrage des Spielers gültig ist, und um sicherzustellen, dass der Spieler nicht einem Platz beitrifft, der zuvor in der Spielsitzung für einen anderen Spieler reserviert war.

Für Spiele, die auf Amazon gehostet werden GameLift, finden Sie unter [client/server Spielinteraktionen mit GameLift Amazon-Servern](#) ein Beispiel dafür, wie diese Art der serverseitigen Validierung implementiert werden kann.

Kundenbeispiel

Beim ersten Betastart AnyCompany von Games wurde festgestellt, dass Spieler ihr Matchmaking-System umgehen, indem sie sich direkt mit Spieleservern verbinden, was zu schwerwiegenden Problemen mit der Wettbewerbsintegrität führte. Als hochrangige Spieler feststellten, dass sie Server-IP-Adressen mit Freunden teilen konnten, begannen sie, das auf Fähigkeiten basierende Matchmaking-System zu umgehen, was dazu führte, dass erfahrene Spieler an Anfängerspielen teilnahmen, was für neue Spieler ein frustrierendes Erlebnis war. AnyCompany Games reagierte darauf mit der Implementierung eines serverseitigen Überprüfungssystems, das für jede Matchmaking-Anfrage eindeutige Sitzungstickets generierte. Das System benötigte sowohl die Spieler IDs - als auch die Matchmaking-Anfrage sowie verifizierte Verbindungsversuche mit dem Backend-Matchmaking-Dienst.

Implementierungsschritte

- Bestätigen Sie Spielerbeitrittsanfragen serverseitig anhand eindeutiger Identifikatoren wie Spielersitzungen und servergenerierter Tickets. IDs
- Bestätigen Sie die Gültigkeit von Verbindungsanfragen mit dem Matchmaking-Service, um unbefugten Zugriff zu blockieren.
- Stellen Sie während des Validierungsprozesses sicher, dass nicht autorisierte Spieler nicht auf reservierte Plätze in Spielsitzungen zugreifen.

GAMESEC03-BP04 Setzen Sie strenge Sicherheitsrichtlinien für Spielerbenutzerkonten durch, indem Sie ein sicheres Passwort verlangen

Wenn ein Spiel Spielern die Möglichkeit bietet, ein Benutzerkonto mit einem Passwort zu erstellen, sollten Sie verlangen, dass sich die Passwörter der Spieler an strenge Richtlinien halten. Amazon Cognito Cognito-Benutzerpools bieten Ihnen beispielsweise die Möglichkeit, [Kennwortanforderungen für Benutzerkonten zu definieren](#). Durch die Einrichtung einer sicheren Passwortrichtlinie können Sie die Konten Ihrer Spieler davor schützen, durch Social Engineering und Brute-Force-Angriffe überholt zu werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Kundenbeispiel

AnyCompany Spiele gerieten in eine Krise, als ihr beliebter Titel aufgrund schwacher Passwortrichtlinien eine Welle von Kontoentführungen erlebte. Spieler, die einfache Passwörter wie „password123“ verwendeten, wurden Opfer automatisierter Brute-Force-Angriffe, die dazu führten, dass Gegenstände verloren gingen und die Spielwährung gefährdet wurde. Um dem entgegenzuwirken, haben AnyCompany Games ihr Anmeldesystem überarbeitet und vorgeschrieben, dass Passwörter, die bisher nicht verwendet wurden, mindestens einen Großbuchstaben, eine Zahl, ein Sonderzeichen und eine Mindestlänge von 15 Zeichen enthalten sollten.

Implementierungsschritte

- Zur Erhöhung der Sicherheit sind strenge Passwortrichtlinien für Spielerkonten erforderlich.
- Verwenden Sie Amazon Cognito Cognito-Benutzerpools, um Passwortanforderungen zu definieren und durchzusetzen.

GAMESEC03-BP05 Bieten Sie Spielern die Möglichkeit, die Multi-Faktor-Authentifizierung (MFA) für ihre Konten einzurichten

Spielerkonten können für böse Akteure eine Bereicherung sein, insbesondere in Spielen, die Spielwährung und Käufe unterstützen. Bieten Sie Spielern aufgrund der weit verbreiteten Verbreitung von Hackerangriffen und Social-Engineering-Angriffen die Möglichkeit, die Sicherheit ihrer Konten zu erhöhen, indem Sie die Multi-Faktor-Authentifizierung (MFA) konfigurieren.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Wenn ein Spieler versucht, mithilfe von MFA auf sein Konto zuzugreifen, wird ein temporärer Code an seine E-Mail-Adresse, Telefonnummer oder eine speziell entwickelte mobile Multi-Faktor-Authentifizierungs-App gesendet. Um sich erfolgreich zu authentifizieren, muss der Spieler den Code dann innerhalb eines begrenzten Zeitraums in das Anmeldesystem eingeben.

MFA kann auch verwendet werden, um Konten zu schützen, die versuchen, sich von einem neuen Standort aus zu authentifizieren, Konten, die vom Spielersupport wegen potenziell bösartiger Aktivitäten gemeldet wurden, und sogar Konten, die sich über einen längeren Zeitraum nicht im Spiel angemeldet haben.

Beispielsweise können Amazon Cognito Cognito-Benutzerpools die [Multi-Faktor-Authentifizierung für Benutzerverzeichnisse konfigurieren](#).

Implementierungsschritte

- Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA), um die Sicherheit der Spielerkonten zu erhöhen.
- Verwenden Sie temporäre Codes, die per E-Mail, Telefon oder MFA-Apps gesendet wurden, um den Kontozugriff zu überprüfen.
- Wenden Sie MFA für neue geografische Standorte, markierte Konten oder Konten mit längerer Inaktivität an.

Zugriffskontrolle

GAMESEC04: Wie blockiert man unbefugten Zugriff auf Spielinhalte?

Moderne Spiele enthalten eine beträchtliche Menge an Inhalten, wie z. B. herunterladbare Inhalte (DLC), was ein wichtiger Aspekt der Spielerbindung und der Monetarisierung von Spielen ist. Die Spieler erwarten ständig neue Charaktere, Level und Herausforderungen, sodass Spieleentwickler mit der ständigen Nachfrage nach neuen Inhalten Schritt halten müssen, um Spieler an sich zu binden. Die Vielfalt und Größe der Inhalte können je nach Art des Spiels und dem Gerät, auf dem das

Spiel gespielt wird, stark variieren. Unabhängig vom System eines Spiels sollten Sie den Inhalt des Spiels vor unbefugtem Zugriff schützen.

Best Practices

- [GAMESEC04-BP01 Beschränken Sie den Zugriff auf herunterladbare Inhalte auf autorisierte Kunden und Benutzer](#)
- [GAMESEC04-BP02 Beschränken Sie den Ursprungszugriff auf autorisierte Content Delivery Networks \(\) CDNs](#)
- [GAMESEC04-BP03 Implementieren Sie geografische Beschränkungen, um unbefugten Zugriff einzuschränken](#)
- [GAMESEC04-BP04 Beschränken Sie den Zugriff auf Inhalte mit DRM-Lösungen \(Digital Rights Management\)](#)

GAMESEC04-BP01 Beschränken Sie den Zugriff auf herunterladbare Inhalte auf autorisierte Kunden und Benutzer

Beschränken Sie den Zugriff auf Spielinhalte durch autorisierte Anwendungen und Clients. Erwägen Sie die Verwendung von Amazon S3 als kostengünstigen und skalierbaren Speicherort für herunterladbare Spielinhalte und Amazon CloudFront, um Spielern eine weltweit leistungsstarke Inhaltsbereitstellung zu bieten. Beide Dienste bieten integrierte Mechanismen zur Beschränkung des Zugriffs auf gespeicherte Daten, z. B. die Beschränkung des Zugriffs auf authentifizierte Benutzer.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Zugriff auf Inhalte gewähren, die in Amazon S3 gespeichert sind

Wenn Sie Zugriff auf Inhalte gewähren müssen, die in S3 gespeichert sind, müssen Sie mehrere Faktoren berücksichtigen. Standardmäßig kann nur derjenige AWS-Konto, der einen S3-Bucket erstellt hat, auf die darin gespeicherten Objekte zugreifen. Um Zugriff auf Ihre internen Anwendungen zu gewähren und in Amazon S3 S3-Buckets gespeicherte Inhalte zu verwalten, verwenden Sie [AWS Identity and Access Management \(IAM\)](#), um Richtlinien zu erstellen, die den entsprechenden Zugriff ermöglichen.

[IAM-Rollen](#) können Verbundbenutzern, Systemen oder Anwendungen zugeordnet werden, die in Services wie Amazon gehostet werden, und containerbasierten Anwendungen EC2 AWS Lambda,

die in Amazon EKS und Amazon ECS gehostet werden. Sie können beispielsweise das AWS SDK verwenden oder AWS CLI Spielinhalte in S3-Buckets veröffentlichen und verwalten. Um diesen Anwendungsfall zu unterstützen, können Sie eine IAM-Rolle mit entsprechendem Zugriff erstellen, um Spielinhalte zu lesen und in Ihre S3-Buckets zu schreiben und sie den EC2 Instanzen zuzuordnen, die Ihre Software und Skripte hosten.

Ressourcenbasierte Richtlinien können für Ihren Bucket und für bestimmte Objekte definiert werden. [S3-Bucket-Richtlinien](#) sind mit einem S3-Bucket verknüpft und können verwendet werden, um den Zugriff auf den Bucket und die darin enthaltenen Objekte einzuschränken sowie den Zugriff auf Ihre Amazon S3 S3-Ressourcen von anderen Konten aus zu gewähren. Beispielsweise können Sie in Szenarien, in denen mehrere Teams oder separate Spieleentwicklungsstudios an denselben Spielinhalten arbeiten und denselben Zugriff auf zentral gehostete Inhalte in Amazon S3 benötigen, eine S3-Bucket-Richtlinie verwenden, um Berechtigungen für den kontenübergreifenden Zugriff auf die S3-Ressourcen zu definieren. Erwägen Sie die Verwendung von [S3-Zugriffspunkten](#), die die Verwaltung des Datenzugriffs auf gemeinsam genutzte Daten vereinfachen können, indem Zugriffspunkte mit Namen und Berechtigungen eingerichtet werden, die für jede Anwendung oder für jede Gruppe von Anwendungen spezifisch sind. Die Amazon S3-Dokumentation enthält zusätzliche [bewährte Methoden für die Zugriffskontrolle in Amazon S3](#).

Granting short-term access to your content

Wenn der Zugriff nur für einen bestimmten, begrenzten Zeitraum erforderlich ist, generieren Sie temporäre URLs Dateien, die Ihnen kurzfristigen Zugriff auf Ihre Inhalte gewähren. Amazon S3 bietet Unterstützung für die Generierung [vorsignierter](#) Objekte URLs, sodass Objekteigentümer zeitlich begrenzten Zugriff auf Objekte in Amazon S3 gewähren können, ohne Ihre Bucket-Richtlinie aktualisieren zu müssen. Auf diese Weise benötigt der Endbenutzer oder die Anwendung, der Zugriff gewährt wird, kein Konto oder keine IAM-Berechtigungen und verwendet stattdessen die vorsignierte URL, um auf den Inhalt zuzugreifen.

Dies ist eine bewährte Methode, die häufig in einer Vielzahl von Anwendungsfällen für Spiele verwendet wird, z. B. um autorisierten Spielern Zugriff auf herunterladbare Inhalte zu gewähren, zu denen sie berechtigt waren, und um temporären Zugriff auf zeitlich begrenzte Spielinhalte zu gewähren. Presigned URLs kann auch verwendet werden, um temporäre Berechtigungen für das Hochladen von Inhalten in einen S3-Bucket bereitzustellen. Du könntest zum Beispiel erwägen, eine vorsignierte URL zu verwenden, um einem Spieler Zugriff auf das Hochladen von Kundenprotokollen zu gewähren, um dein Support-Team bei der Behebung eines Spieler-Supportfalls zu unterstützen.

Verwenden Sie ein Content Delivery Network, um Zugriff auf Ihre Inhalte zu gewähren

Während Ihre Anwendungen, Spieleentwickler, Künstler und anderes Personal zu Entwicklungs- und Verwaltungszwecken möglicherweise direkten Zugriff auf die Inhalte in S3-Buckets benötigen, verwenden Sie ein Content Delivery Network, um den Zugriff auf Inhalte zu ermöglichen, die Spielern oder anderen Benutzern über das Internet öffentlich zugänglich sind. Dieser Ansatz verbessert die Download-Leistung und senkt die Kosten, indem häufig aufgerufene Inhalte zwischengespeichert werden. Amazon CloudFront kann Ihre Inhalte weltweit verteilen, indem es sie zwischenspeichert und Ihren Spielern näher bringt und gleichzeitig die Belastung des Download-Ursprungs Ihres Spiels reduziert, z. B. Amazon S3.

Anstatt Ihre öffentlichen Inhalte direkt aus S3-Buckets bereitzustellen, empfiehlt es sich, diese Inhalte privat zu halten und sie öffentlich bereitzustellen, indem Sie CloudFront CloudFront kann so konfiguriert werden, dass Spieler entweder [signierte URLs](#) oder [signierte Cookies](#) verwenden müssen, um auf deine privaten Inhalte zuzugreifen (z. B. ein neues Spiel nur für kostenpflichtige Spieler). Anschließend können Sie Ihre Anwendung so entwickeln, dass entweder signierte Benutzer erstellt und URLs an authentifizierte Benutzer verteilt werden oder dass Set-Cookie-Header gesendet werden, die signierte Cookies für authentifizierte Benutzer setzen. Wenn Sie signierte URLs oder signierte Cookies erstellen, um den Zugriff auf Ihre Dateien zu kontrollieren, können Sie ein Enddatum und eine Endzeit angeben. Danach sind die URL und die Cookies nicht mehr gültig.

Optional können Sie auch die IP-Adresse oder den Adressbereich der Computer angeben, die für den Zugriff auf Ihre Inhalte verwendet werden können. Dies ist nützlich, wenn Sie den Zugriff auf bestimmte Spieleentwicklungsstudio-Partner oder Vertragsnetzwerke einschränken möchten. Verwenden Sie signierte Cookies, wenn Sie Zugriff auf mehrere eingeschränkte Dateien gewähren möchten oder wenn Sie Ihre aktuellen Dateien nicht ändern möchten URLs. Verwenden Sie signierte Dateien, URLs wenn Sie den Zugriff auf einzelne Dateien einschränken möchten oder wenn Ihre Benutzer einen Client verwenden, der keine Cookies unterstützt. Signierte Cookies haben URLs Vorrang vor signierten Cookies.

Implementierungsschritte

- Verwenden Sie IAM-Rollen und Bucket-Richtlinien, um internen Anwendungen, Teams oder kontenübergreifenden Szenarien angemessenen Zugriff auf S3-Buckets zu gewähren.
- Generieren Sie Objekte, URLs die so konzipiert sind, dass sie kurzfristigen Zugriff auf S3-Objekte gewähren. Diese eignen sich für herunterladbare Inhalte oder temporäre Uploads wie Client-Logs.
- Verwenden Sie Amazon CloudFront mit signierten Cookies URLs oder Cookies, um privaten Inhalten authentifizierte Benutzern sicherer bereitzustellen

GAMESEC04-BP02 Beschränken Sie den Ursprungszugriff auf autorisierte Content Delivery Networks () CDNs

Blockieren Sie Benutzer daran, Ihre Content Delivery Networks zu umgehen, um direkt auf Inhalte von Ihrem Ursprung zuzugreifen, z. B. Ihre Amazon S3 S3-Buckets. Es ist wichtig, dass Sie den Zugriff auf Ihre Herkunft nur auf Ihre autorisierten Personen beschränken CDNs, um die Kosten für die Datenübertragung zu reduzieren, die durch die unnötige Bereitstellung von Inhalten außerhalb des Ursprungs entstehen. Es verbessert auch Ihre Sicherheitslage, indem der öffentliche Zugriff auf Ihre ursprünglichen Inhalte über denselben Zugangspunkt geleitet wird. Dort können Sie Edge-Sicherheitskontrollen wie AWS WAF Layer-7-Filterung, Injektion und Überprüfung sicherheitsrelevanter HTTP-Anforderungsparameter und verteilte Denial-of-Service (DDoS) - Schutzmaßnahmen einsetzen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Um diese Kontrollen für einen Amazon S3 S3-Ursprung zu implementieren, können Sie eine [Amazon CloudFront Origin Access Identity \(OAI\)](#) verwenden, die überprüft, ob Anfragen an Ihre S3-Objekte von Ihrer CloudFront Distribution stammen. Stellen Sie eine AWS WAF Verbindung zu Ihrer CloudFront Distribution her, um Layer-7-Filterung bereitzustellen. Wenn Sie jedoch Inhalte von zusätzlichen Quellen bereitstellen CDNs, können Sie das CDN so konfigurieren, dass ein oder mehrere benutzerdefinierte HTTP-Header in ursprüngliche Anfragen eingefügt werden. Anhand dieser können Sie überprüfen, ob der eingehende Datenverkehr von AWS WAF Ihrem autorisierten CDN-Anbieter stammt.

Dieser Ansatz ist auch nützlich, um zu verhindern, dass Benutzer Ihre CDN-Anbieter umgehen, wenn Ihr Ursprung hinter einem [Application Load Balancer](#) (ALB) gehostet wird. ALBs kann für Layer-7-Schutzmaßnahmen verwendet werden. AWS WAF Sie können so konfigurieren AWS WAF , dass ein benutzerdefinierter HTTP-Header eingefügt wird, der von Ihrem ALB überprüft wird, um den eingehenden Datenverkehr zum Load Balancer zu verarbeiten und zu überprüfen. AWS WAF

Kundenbeispiel

AnyCompany Games implementiert Beschränkungen für den Zugriff auf die Herkunft, um ihre Spielressourcen, herunterladbaren Inhalte und Patch-Dateien vor unberechtigtem direktem Zugriff zu schützen, der es Spielern ermöglichen könnte, Sicherheitsüberprüfungen zu umgehen oder Premium-Inhalte ohne ordnungsgemäße Authentifizierung zu erhalten. Dieser Ansatz ermöglicht es ihnen,

die Zugriffsmuster auf Inhalte von einem zentralen Punkt aus zu überwachen, sodass sie leicht verdächtiges Download-Verhalten erkennen können, das auf koordinierte Angriffe oder die unbefugte Weiterverbreitung von Inhalten hinweisen könnte.

Implementierungsschritte

- Verwenden Sie Amazon CloudFront Origin Access Identity (OAI), um den direkten Zugriff auf S3-Objekte einzuschränken
- Stellen Sie eine Verbindung AWS WAF mit CloudFront oder ALB her, um Layer-7-Filterung bereitzustellen und sich vor DDoS-Angriffen und böswilligen Anfragen zu schützen.
- Konfigurieren Sie benutzerdefinierte HTTP-Header in Cloudfront, um sicherzustellen, dass der eingehende Datenverkehr aus autorisierten Quellen stammt.

GAMESEC04-BP03 Implementieren Sie geografische Beschränkungen, um unbefugten Zugriff einzuschränken

Wenn ein Spieler Ihre Inhalte anfordert, CloudFront stellt Amazon die angeforderten Inhalte vom nächstgelegenen Edge-Standort aus bereit, unabhängig davon, wo sich der Player befindet. Es kann jedoch Szenarien geben, in denen Sie einschränken müssen, wie Nutzer in bestimmten Teilen der Welt auf Ihre Inhalte zugreifen können. Möglicherweise haben Sie eine fortlaufende Bereitstellungsstrategie, bei der Inhalte phasenweise veröffentlicht werden, oder Sie müssen sich möglicherweise an länderspezifische Zugriffskontrollen halten. country-by-country

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Du kannst [geografische Beschränkungen](#), auch als Geoblocking bezeichnet, verwenden, um Spieler an bestimmten geografischen Standorten daran zu hindern, auf Inhalte zuzugreifen, die du über eine CloudFront Distribution verbreitest. Mit dieser Funktion können Sie den Zugriff auf Dateien einschränken, die mit einer Distribution verknüpft sind, und den Zugriff auf Landesebene einschränken. Alternativ können Sie einen Geolokalisierungsdienst eines Drittanbieters verwenden, um den Zugriff auf eine Teilmenge der Dateien zu beschränken, die mit einer Distribution verknüpft sind, oder um den Zugriff auf eine feinere Granularität als auf Landesebene einzuschränken.

Durch CloudFront geografische Beschränkungen kannst du deinen Spielern erlauben, nur auf deine Inhalte zuzugreifen, wenn sie sich in einem der Länder befinden, die auf einer Liste zugelassener

Länder stehen. Du kannst deine Spieler auch daran hindern, auf deine Inhalte zuzugreifen, wenn sie sich in einem der Länder befinden, die auf einer Sperrliste der Länder stehen, die gesperrt sind. Wenn eine Anfrage von einem blockierten geografischen Standort eingeht, CloudFront wird dem Spieler der HTTP-Statuscode 403 Forbidden zurückgegeben. Es ist wichtig zu beachten, dass dies bei nicht sensiblen Inhalten gut funktioniert und nicht als eigenständiger Schutz für personenbezogene Daten oder sensible Spielartefakte verwendet werden sollte.

Implementierungsschritte

- Verwende CloudFront geografische Beschränkungen, um den Zugriff auf Inhalte auf der Grundlage von Zulassungs- oder Verweigerungslisten auf Landesebene zuzulassen oder zu verweigern.
- Gib den HTTP-Statuscode 403 Forbidden für Anfragen zurück, die von blockierten geografischen Standorten stammen.
- Vermeiden Sie es, sich ausschließlich auf geografische Beschränkungen zu verlassen, um vertrauliche Inhalte oder personenbezogene Daten zu schützen

GAMESEC04-BP04 Beschränken Sie den Zugriff auf Inhalte mit DRM-Lösungen (Digital Rights Management)

Erwägen Sie, den Zugriff auf Ihre Spielinhalte einzuschränken, indem Sie starke Verschlüsselungstools wie eine [DRM-Lösung \(Digital Rights Management\)](#) verwenden. Diese Art von Lösung kann verwendet werden, um Ihre privaten Inhalte zu verschlüsseln und die Entschlüsselungsschlüssel an autorisierte Spieler zu verteilen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

DRM-Lösungen werden in Situationen empfohlen, in denen Sie es Spielern ermöglichen möchten, Spielinhalte frühzeitig herunterzuladen, aber Sie möchten nicht, dass sie bis zu einem vorher festgelegten Zeitpunkt auf die Inhalte zugreifen oder diese abspielen können. Dies ist beispielsweise häufig in Situationen der Fall, in denen Spieler ein Spiel vorbestellen und ihren Spielclient so konfigurieren dürfen, dass er die verschlüsselten Dateien automatisch frühzeitig herunterlädt. Diese Strategie stellt sicher, dass das Spiel heruntergeladen wurde und spielbereit ist, sobald das Spiel offiziell veröffentlicht wurde. Nach der Veröffentlichung des Spiels kann der Spielclient des Spielers Entschlüsselungsschlüssel von der DRM-Backend-Lösung anfordern, damit er die zuvor heruntergeladenen Dateien entschlüsseln und mit dem Spielen des Spiels beginnen kann.

DRM-Systeme werden auch verwendet, um die unbefugte Weiterverbreitung und Manipulation von Spielen zu verhindern, nachdem sie von einem autorisierten Spieler heruntergeladen und installiert wurden. DRM-Systeme erfordern eine Integration mit der Quelle, um Verschlüsselungsschlüssel auszutauschen und Spieler zum Abrufen des Entschlüsselungsschlüssels zu autorisieren. Kommerzielle DRM-Anbieter bieten eine Reihe von Lösungen mit Funktionen und Unterstützung für verschiedene Geräte.

Implementierungsschritte

- Verwenden Sie DRM-Lösungen, um private Spielinhalte zu verschlüsseln und Entschlüsselungsschlüssel an autorisierte Spieler zu verteilen.
- Ermöglichen Sie den Vorab-Download verschlüsselter Dateien für vorbestellte Spiele und entsperren Sie den Zugriff mit Entschlüsselungsschlüsseln zum Zeitpunkt der Veröffentlichung.
- Integrieren Sie DRM-Systeme mit dem Ursprung, um die Verschlüsselungsschlüssel zu verwalten und die unbefugte Weiterverbreitung oder Manipulation von Inhalten zu verhindern.

Erkennung

GAMESEC05: Wie überwachst und analysierst du das Nutzungsverhalten der Spieler in deinem Spiel?

Die Überwachung und Analyse des Nutzungsverhaltens von Spielern ist für Spielestudios unerlässlich, da Sie damit Sicherheitsbedrohungen, Cheats und andere Formen missbräuchlichen Verhaltens erkennen können, die die Integrität und die Sicherheit der Spieler gefährden könnten. Indem Sie Muster wie ungewöhnliche Fortschrittsraten, ungewöhnliche Transaktionen im Spiel oder verdächtiges Kommunikationsverhalten verfolgen, können Sie potenzielle Cheater, betrügerische Konten oder koordinierte Bedrohungen identifizieren, bevor sie das Spielerlebnis erheblich beeinträchtigen.

Best Practices

- [GAMESEC05-BP01 Implementieren Sie eine umfassende Datenerfassungsstrategie zur Überwachung des Spielerverhaltens](#)
- [GAMESEC05-BP02 Sammeln, speichern und analysieren Sie Nutzungsprotokolle von Spielern, um unangemessenes Verhalten zu erkennen](#)

GAMESEC05-BP01 Implementieren Sie eine umfassende Datenerfassungsstrategie zur Überwachung des Spielerverhaltens

Implementieren Sie eine umfassende Strategie zur Datenerfassung und -analyse, um ein positives Spielerlebnis aufrechtzuerhalten. Das Erfassen, Speichern und Analysieren relevanter Daten bietet Einblicke in die Art und Weise, wie Spieler mit den Funktionen Ihres Spiels und miteinander interagieren. Dieser datengestützte Ansatz kann als Entscheidungshilfe dienen, das Engagement und die Bindung der Spieler verbessern, Monetarisierungsstrategien optimieren und letztendlich das Spielerlebnis insgesamt verbessern.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Implementieren Sie Datenerfassungssysteme, um relevante Spieleraktionen wie Spielsitzungen, Fortschritte, Erfolge, Käufe, Interaktionen mit Spielelementen und soziale Aktivitäten zu erfassen und zu protokollieren. Sammeln Sie serverseitige Daten wie Serverlast, Netzwerkverkehr und Fehlerprotokolle, um die technische Leistung zu überwachen und potenzielle Probleme zu identifizieren. Sammeln Sie Feedback von Spielern mithilfe von Umfragen, Foren, Support-Tickets und Social-Media-Kanälen, um mehr über ihre Erfahrungen und Vorlieben zu erfahren.

Richten Sie beim Speichern Ihrer Spieldaten ein zentrales Data Warehouse oder einen Data Lake ein, um die gesammelten Daten zu speichern und zu organisieren, und implementieren Sie Pipelines für die Datenbereinigung, -transformation und -aggregation, um die Daten für eine effiziente Analyse vorzubereiten.

Analysieren Sie die Daten nach dem Speichern, um mithilfe von Datenvisualisierungstools Einblicke in die Spielerbindung und -abwanderung, Monetarisierungsstrategien und die Nutzung von Funktionen zu gewinnen.

Implementierungsschritte

- Erfassen und protokollieren Sie Spieleraktionen, serverseitige Metriken und Feedback, um Interaktionen und technische Leistung zu überwachen.
- Verwenden Sie ein zentrales Data Warehouse wie Amazon Redshift oder S3 Data Lake, um Spieldaten für die Analyse zu speichern, zu bereinigen, zu transformieren und zu organisieren.
- Analysieren Sie die gesammelten Daten mit Visualisierungstools wie Amazon Quicksight, um Einblicke in die Spielerbindung, Monetarisierung und Nutzung von Funktionen zu erhalten.

GAMESEC05-BP02 Sammeln, speichern und analysieren Sie Nutzungsprotokolle von Spielern, um unangemessenes Verhalten zu erkennen

Instrumentiere dein Spiel, um Logs zu sammeln, um zu verstehen, wie Spieler die Funktionen deines Spiels nutzen und wie sie mit anderen Spielern interagieren. Sie können dann unautorisierte Aktivitäten blockieren, die das Spielerlebnis beeinträchtigen können.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

[Senden Sie strukturierte Protokollereignisse an die Game Analytics-Pipeline, indem Sie eine Protokollierungslösung wie Amazon CloudWatch Logs oder Amazon OpenSearch Service oder eine Lösung von einem AWS Partner wie Datadog, Sumo Logic, New Relic, Honeycomb.io oder Splunk verwenden.](#) Strukturieren Sie diese Spieler-Nutzungsprotokolle so, dass anhand ihrer Hilfe festgestellt werden kann, wann bestimmte Aktionen von Spielern untersucht werden müssen.

Nachdem Sie die Daten erfasst haben, sollten Sie die Implementierung von Tools zur Erkennung von unangemessenem Nutzungsverhalten in Betracht ziehen. Wenn Ihr Spiel beispielsweise über soziale Funktionen wie Spielernachrichten im Spiel, Sprachchat oder Online-Foren verfügt, speichern Sie die Protokolle dieser Spielerinteraktionen in einem Format, das zu Moderationszwecken analysiert werden kann.

Konfigurieren Sie die Voice-Chat-Funktion Ihres Spiels, um Aufzeichnungen nach Amazon S3 zu exportieren, und verwenden Sie [Amazon Transcribe](#), um die Audiosprache in ein Textformat zu konvertieren, das zur Verarbeitung gespeichert werden kann. Alternativ können Sie Streaming-Transkription in Echtzeit durchführen, indem Sie Ihren Game-Backend-Voice-Chat-Dienst direkt in die Transcribe-API integrieren, um Streaming-Audio in Echtzeit zu [Transcribe](#). Moderationsteams können den Inhalt manuell überprüfen. Sobald der Inhalt in einem Standardformat vorliegt, können Sie auch KI/ML-Dienste verwenden AWS, um die Moderation automatisch durchzuführen. [Amazon Comprehend](#) kann zur Verarbeitung natürlicher Sprache (NLP) verwendet werden, um Informationen aus dem unstrukturierten Text aufzudecken. Dadurch können die Konversationen nach relevanten Themen klassifiziert und organisiert und unangemessenes Verhalten wie Obszönitäten identifiziert werden.

Implementierungsschritte

- Sammeln, speichern und analysieren Sie Nutzungsprotokolle von Spielern.

- Nutzen Sie AWS Dienste für künstliche Intelligenz und maschinelles Lernen, um Ihre Spieler-Nutzungsprotokolle effizienter zu überprüfen und Einblicke in sie zu gewinnen.

Schutz der Infrastruktur

Im Whitepaper Well-Architected Framework finden Sie bewährte Methoden zum [Infrastrukturschutz](#) für die Sicherheit, die für Spiele-Workloads gelten.

GAMESEC06: Wie überwachen und reagieren Sie auf Infrastrukturbedrohungen?

Die Überwachung und Reaktion auf Infrastrukturbedrohungen ist für Spielestudios von entscheidender Bedeutung, da ihre Infrastruktur das Rückgrat darstellt, das Millionen von gleichzeitigen Spielern unterstützt, Echtgeldtransaktionen verarbeitet und wertvolle Spielerdaten und firmeneigene Spielinhalte speichert. Für Spielestudios ist es von entscheidender Bedeutung, Überwachungssysteme und Verfahren zur Reaktion auf Vorfälle zu implementieren, mit denen die Integrität sowohl der Spielererlebnisse als auch der Geschäftsabläufe gewahrt werden kann.

Best Practices

- [GAMESEC06-BP01 Verwenden Sie Tools, um Bedrohungen für Ihre Infrastruktur zu erkennen und darauf zu reagieren](#)
- [GAMESEC06-BP02 Verwenden Sie Tools für künstliche Intelligenz und maschinelles Lernen, um Aspekte Ihrer Infrastrukturschutzstrategie zu automatisieren](#)
- [GAMESEC06-BP03 Nutzen Sie Erkenntnisse aus Protokollen auf Systemebene, um Ihre Infrastrukturschutzstrategie kontinuierlich zu verbessern](#)

GAMESEC06-BP01 Verwenden Sie Tools, um Bedrohungen für Ihre Infrastruktur zu erkennen und darauf zu reagieren

Um Ihre AWS Umgebung kontinuierlich auf böswillige Aktivitäten und unbefugtes Verhalten zu überwachen, sollten Sie [Amazon](#) in Betracht ziehen GuardDuty. GuardDuty identifiziert Bedrohungen durch die Überwachung des Kontoverhaltens, der Netzwerkaktivitäten und der Datenzugriffsmuster in Ihrer Umgebung. Es analysiert Ereignisse aus mehreren Datenquellen, wie z. B. CloudTrail Ereignisprotokollen, Amazon VPC Flow Logs und DNS-Protokollen, auf

potenzielle Bedrohungen. Durch die Integration mit Amazon CloudWatch Events und Lambda können GuardDuty Warnmeldungen zur weiteren Analyse automatisch an die zuständigen Sicherheitsteams weitergeleitet werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

[AWS Security Hub CSPM](#) bietet einen umfassenden Überblick über Ihren Sicherheitsstatus in Ihrer Umgebung AWS und überprüft Ihre Umgebung anhand von Industriestandards und Best Practices. Security Hub CSPM sammelt Sicherheitsdaten von Across AWS-Konten, Services und unterstützten Produkten von Drittanbietern, analysiert Ihre Sicherheitstrends und identifiziert die Sicherheitsprobleme mit der höchsten Priorität. Die [GuardDutyAmazon-Integration mit Security Hub CSPM](#) ermöglicht es Ihnen, Ergebnisse von an Security Hub CSPM GuardDuty zu senden. Security Hub CSPM kann diese Ergebnisse dann in die Analyse Ihres Sicherheitsstatus einbeziehen.

Es ist üblich, dass böswillige Akteure Bots einsetzen, um Konten zu übernehmen und in Spielen zu cheaten. [WAF Bot Control](#) bietet Ihnen Transparenz und Kontrolle über den allgemeinen und allgegenwärtigen Bot-Verkehr, der überschüssige Ressourcen verbrauchen, Metriken verzerren, Ausfallzeiten verursachen oder andere unerwünschte Aktivitäten ausführen kann.

Ransomware ist bösartiger Code, der darauf abzielt, sich unbefugten Zugriff auf Systeme und Datensätze zu verschaffen und diese Daten zu verschlüsseln, um den Zugriff durch legitime Spieler zu blockieren. Nachdem Ransomware Spieler von ihren Systemen ausgesperrt und ihre sensiblen Daten verschlüsselt hat, verlangen Cyberkriminelle ein Lösegeld, bevor sie einen Entschlüsselungsschlüssel zum Entsperren der Daten bereitstellen. Organizations können durch ein böswilliges Ereignis vollständig zum Erliegen gebracht werden, was zu erheblichen Kosten und einem Verlust der Geschäftsproduktivität führen kann. Unter [Schützen Ihrer AWS Cloud Umgebung vor Ransomware](#) finden Sie bewährte Methoden, die Sie anwenden können, um Ihre Fähigkeit zur Bekämpfung von Ransomware vor, während und nach einem Vorfall zu verbessern.

Ihr Spiel bietet Spielern möglicherweise die Möglichkeit, Spielersupport-Mitarbeiter über ein Call Center wie [Amazon Connect](#) oder Chat-Bots über Amazon Lex zu kontaktieren. Amazon Connect bietet Unterstützung für die [Überwachung von Live-Konversationen und aufgezeichneten Gesprächen](#). Um Interaktionen zwischen Spielern und mit Amazon Lex erstellten Chat-Bots zur Spielerunterstützung zu analysieren, können Sie die [Konversationsprotokolle](#) dieser Interaktionen in Amazon CloudWatch Logs speichern, die nach Amazon S3 exportiert und wie zuvor beschrieben analysiert werden können.

Führen Sie abschließend Penetrationstests als Teil Ihrer Infrastrukturschutzstrategie durch. Unabhängig davon, ob Sie diese Bewertungen intern oder über einen AWS Partner durchführen, halten Sie sich an die [Richtlinien des AWS Kundensupports für Penetrationstests](#).

Implementierungsschritte

- Verwenden Sie Amazon, GuardDuty um das Kontoverhalten, die Netzwerkaktivität und die Datenzugriffsmuster im Hinblick auf Bedrohungen zu überwachen, und integrieren Sie es mit Security Hub CSPM, um eine einheitliche Sicherheitsansicht zu erhalten.
- Implementieren Sie AWS WAF Bot Control, um Bot-Traffic zu erkennen und einzudämmen, der Ressourcen und Spielerlebnisse schaden kann.
- Führen Sie regelmäßig Penetrationstests durch und halten Sie sich dabei an die Richtlinien für den AWS Kundensupport, um Ihr Sicherheitsniveau zu bewerten und zu verbessern.

GAMESEC06-BP02 Verwenden Sie Tools für künstliche Intelligenz und maschinelles Lernen, um Aspekte Ihrer Infrastrukturschutzstrategie zu automatisieren

[Amazon Lookout for Metrics](#) verwendet maschinelles Lernen, um Anomalien in Ihren Geschäfts- und Betriebsdaten automatisch zu erkennen und zu diagnostizieren, und überwacht die Kennzahlen, die für Ihr Unternehmen am wichtigsten sind, schneller und genauer. Der Service macht es auch einfach, die Grundursache von Anomalien zu diagnostizieren, wie z. B. plötzliche Einbrüche bei Umsatz, Anmeldungen, Transaktionen oder Aufbewahrung. Für die Einrichtung müssen Spieleentwickler keine ML-Erfahrung haben und es kann eine Verbindung zu beliebigen Datenquellen wie Amazon S3, Amazon, Amazon CloudWatch RDS, Amazon Redshift sowie vielen SaaS-Anwendungen hergestellt werden. Sie können [Amazon Lookout for Metrics beispielsweise in die Game Analytics-Pipeline und andere Datenquellen integrieren, um mit](#) der Verhaltensanalyse zur Erkennung von Anomalien zu beginnen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Alternativ können Sie mithilfe von [Amazon SageMaker AI](#) ein benutzerdefiniertes Modell für maschinelles Lernen erstellen, trainieren und hosten, um Anwendungsfälle wie Inhaltsmoderation, Toxizitätserkennung, Cheat-Erkennung, Betrugserkennung und mehr zu adressieren.

Kundenbeispiel

AnyCompany Games verwendet Amazon Lookout for Metrics, um automatisch ungewöhnliche Muster bei der Serverleistung, bei Anmeldeversuchen von Spielern oder beim Transaktionsvolumen zu erkennen, die auf Bedrohungen durch böswillige Akteure hinweisen könnten. Darüber hinaus haben sie Amazon SageMaker AI verwendet, um maßgeschneiderte Modelle für maschinelles Lernen zu entwickeln, die kontinuierlich Netzwerkverkehrsmuster und Spielerverhalten analysieren, um koordinierte Bedrohungen zu identifizieren, z. B. Bot-Netzwerke, die versuchen, ihre virtuelle Wirtschaft auszunutzen.

Dieser automatisierte Ansatz ermöglicht es dem Sicherheitsteam, sich auf die Untersuchung und Reaktion auf echte Bedrohungen zu konzentrieren, anstatt Tausende von Kennzahlen manuell zu überwachen. Gleichzeitig wird sichergestellt, dass neu auftretende Bedrohungsmuster erkannt und behoben werden, bevor sie die Verfügbarkeit von Spielen oder die Sicherheit der Spieler erheblich beeinträchtigen können.

Implementierungsschritte

- Verwenden Sie Amazon Lookout for Metrics, um Anomalien in wichtigen Geschäfts- und Betriebsdaten automatisch zu erkennen und zu diagnostizieren
- Integrieren Sie Amazon Lookout for Metrics mit Datenquellen wie der Game Analytics Pipeline oder Amazon S3 oder CloudWatch zur Überwachung von Kennzahlen wie Umsatz, Anmeldungen und Kundenbindung.
- Verwenden Sie Amazon SageMaker AI, um benutzerdefinierte Modelle für maschinelles Lernen für fortgeschrittene Anwendungsfälle wie Cheat-Erkennung, Betrugsprävention und Inhaltsmoderation zu erstellen, zu trainieren und zu hosten.

GAMESEC06-BP03 Nutzen Sie Erkenntnisse aus Protokollen auf Systemebene, um Ihre Infrastrukturschutzstrategie kontinuierlich zu verbessern

[Erfassen und speichern Sie Protokolle auf Systemebene von relevanten Diensten, wie z. B. S3-Serverzugriffsprotokolle, CloudFrontZugriffsprotokolle und ALB-Zugriffsprotokolle.](#) Diese Protokolle können in einem S3-Bucket in Ihrem Konto gespeichert werden und sind nützlich, um Ihre Spielernutzungsinformationen innerhalb des Spiels mit Informationen auf Systemebene zu verknüpfen, einschließlich Verbindungsdetails wie IP-Adressen, Anforderungsheadern und relevanter Bearbeitung und Filterung von Anfragen, die Sie möglicherweise in Ihrem Spiel-Backend konfiguriert haben. Sie können diese Protokolle an dieselben Protokollierungslösungen senden, die bereits

erwähnt wurden, und Sie können [sie mithilfe von SQL-Abfragen mit Amazon Athena analysieren](#), ohne dass die Protokolle aus Amazon S3 verschoben werden müssen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

[Access Analyzer for S3](#) ist eine Funktion, die Ihre Bucket-Zugriffsrichtlinien überwacht und sicherstellt, dass die Richtlinien nur den beabsichtigten Zugriff auf Ihre Amazon S3 S3-Ressourcen gewähren. Access Analyzer for S3 bewertet Ihre Bucket-Zugriffsrichtlinien und ermöglicht es Ihnen, Buckets mit potenziell unbeabsichtigtem Zugriff zu erkennen und schnell zu beheben.

Implementierungsschritte

- Nutzen Sie AWS Dienste zur Erkennung von Bedrohungen und zur Reaktion auf Vorfälle, um Aspekte Ihrer Infrastrukturschutzstrategie zu automatisieren.
- Gewinnen Sie mithilfe von Protokollen und AWS Diensten auf Systemebene für künstliche Intelligenz und maschinelles Lernen Einblicke in den Schutz Ihrer Infrastruktur.

Datenschutz

Denken Sie bei der Entwicklung und Architektur Ihres Spiels darüber nach, welche Art von Daten Ihr Studio sammelt und wie Sie sich für den Schutz dieser Daten entschieden haben. Zu den Themen, die es im Zusammenhang mit diesem Sicherheitsaspekt zu untersuchen gilt, gehören:

- Wie haben Sie sich entschieden, Ihre Daten zu identifizieren und zu klassifizieren
- Wie schützen Sie Daten im Ruhezustand
- Wie schützen Sie Daten bei der Übertragung

Für Games Lens gibt es keine bewährten Datenschutzpraktiken. Bewährte Verfahren zum [Datenschutz](#) aus Sicherheitsgründen finden Sie im Whitepaper Well-Architected Framework.

Vorfallreaktion

GAMESEC07: Wie definierst und setzt du Richtlinien durch, um auf Fehlverhalten und missbräuhliches Verhalten von Spielern zu reagieren?

Fehlverhalten und missbräuchliches Verhalten von Spielern können das Spielerlebnis erheblich beeinträchtigen. Schlechtes Spielerverhalten kann legitime Spieler vertreiben, was zu einer geringeren Spielerbindung, geringeren Einnahmen aus Käufen im Spiel und negativen Bewertungen führt, die den Ruf eines Spiels und future Verkäufe schädigen können.

Definieren Sie Richtlinien, die positive Aktionen Ihrer Spieler fördern, und legen Sie fest, wie Sie diese Richtlinien durchsetzen.

Best Practices

- [GAMESEC07-BP01 Implementieren Sie einen Plan zur Reaktion auf Vorfälle, um mit schlechten Akteuren und missbräuchlichem Verhalten umzugehen](#)
- [GAMESEC07-BP02 Konten sperren, die mit schlechten Schauspielern in Verbindung gebracht werden](#)

GAMESEC07-BP01 Implementieren Sie einen Plan zur Reaktion auf Vorfälle, um mit schlechten Akteuren und missbräuchlichem Verhalten umzugehen

Erstelle einen Aktionsplan, um auf schlechte Schauspieler und missbräuchliches Verhalten in deinem Spiel zu reagieren.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Berücksichtigen Sie Faktoren wie den Zeitpunkt der vorübergehenden oder dauerhaften Sperrung von Spielern und die Dauer der Deaktivierung der Zugangsdaten für vorübergehend gesperrte Spieler.

Kundenbeispiel

AnyCompany Games führt ein abgestuftes System zur Reaktion auf Vorfälle ein, bei dem geringfügige Verstöße wie unangemessene Chat-Nachrichten zu automatischen 24-stündigen Kontosperrungen führen, während schwerwiegendere Verstöße wie Cheaten oder Belästigung sofortige 7-tägige Sperrungen mit obligatorischer Überprüfung durch menschliche Moderatoren nach sich ziehen.

Darüber hinaus führt AnyCompany Games Eskalationsverfahren ein, bei denen Wiederholungstäter mit immer längeren Sperrungen rechnen müssen. Sie führen Einspruchsverfahren ein, die es Spielern

ermöglichen, fälschlicherweise gemeldete Aktionen anzufechten und gleichzeitig die Sicherheit durch Anforderungen zur Identitätsprüfung zu wahren.

GAMESEC07-BP02 Konten sperren, die mit schlechten Schauspielern in Verbindung gebracht werden

Wenn nicht eingedämmt wird, kann sich missbräuchliches Verhalten in einem Spiel weiterhin negativ auf das Spielerlebnis anderer auswirken und sollte so schnell wie möglich gemildert werden. Implementieren Sie ein Verfahren, um böswilligen Akteuren, die nachweislich gegen Ihre Nutzungsbedingungen verstoßen, Verbote oder andere Beschränkungen aufzuerlegen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

In der Regel werden die Regeln und der Bewertungsprozess zur Bestimmung der Umstände, unter denen diese Art von Beschränkungen verhängt werden, von Mitarbeitern festgelegt, z. B. von einem Spielergemeinschaftsteam oder einem Vertrauens- und Sicherheitsteam innerhalb Ihrer Organisation. Nachdem Sie die böswilligen Akteure gekennzeichnet haben, führen Sie einen vorab festgelegten Arbeitsablauf durch, um gegen die identifizierten Spieler vorzugehen.

Zum Beispiel können [AWS Lambda](#) Funktionen verwendet werden, [AWS Step Functions](#) um einen automatisierten Workflow auszuführen, der eine Reihe von Spielerkonten als Eingabe akzeptiert. Der Workflow aktualisiert dann Einträge in einer [Amazon DynamoDB-Tabelle](#) namens Bans, die Details über das Spielerkonto, den Grund und die Dauer der Sperre enthalten kann.

Abhängig von der Gestaltung Ihres Spiel- und Kontoverwaltungssystems und der Art des Missbrauchs, dem Sie durch böswillige Akteure ausgesetzt sind, sollten Sie ein von Ihrem Kontoverwaltungssystem getrenntes Aufzeichnungssystem für Sperren einrichten. Möglicherweise möchten Sie das Konto des Spielers nicht in Ihrem Kontoverwaltungssystem deaktivieren, sondern stattdessen einfach seine Fähigkeit ausschalten, Ihr Spiel zu spielen. Dies kann in Situationen nützlich sein, in denen die Kontodaten des Spielers für den Zugriff auf mehrere Spiele mit unterschiedlichen Nutzungsbedingungen oder Richtlinien verwendet werden.

Implementierungsschritte

- Definieren und durchsetzen Sie Richtlinien für die Reaktion auf missbräuchliches Verhalten bösartiger Akteure.
- Verwenden Sie AWS Dienste, um Ihre Reaktionen auf böswillige Akteure zu automatisieren.

Ressourcen

- [AWS Technischer Leitfaden zur Reaktion auf Sicherheitsvorfälle](#)
- [AWS Blog Machine Learning: Mit Amazon Rekognition Face Liveness echte und echte Nutzer erkennen und böswillige Akteure abschrecken](#)
- [AWS Lösungen für Spiele: Community Health](#)

Anwendungssicherheit

GAMESEC08: Wie sichern Sie Ihre CI/CD Pipeline?

Eine CI/CD Spieleentwicklungspipeline besteht in der Regel aus hochverfügbaren Quellcodeverwaltungsservern und Speicher, Rechenressourcen für die Ausführung Ihrer Builds und Software zur Durchführung automatisierter Tests sowie aus der richtigen Netzwerkkonnektivität Ihrer Entwicklungscomputer. Die Sicherung Ihrer CI/CD Pipeline ist wichtig, um vertrauliche Informationen zu schützen, die Codeintegrität zu wahren und vertrauenswürdige Versionen aufrechtzuerhalten. Die Einbettung von Governance und Leitplanken ermöglicht Entwicklern Flexibilität bei gleichzeitiger Beibehaltung guter Sicherheitspraktiken.

Da Spiele häufig die Zahlungsabwicklung übernehmen, persönliche Daten speichern und virtuelle Ökonomien unterhalten, die echtes Geld wert sind, kann eine Sicherheitsverletzung im Entwicklungsprozess zu erheblichen finanziellen Verlusten, behördlichen Sanktionen und einem Verlust des Vertrauens der Spieler führen.

Durch die Integration von Sicherheitsvorkehrungen behalten Unternehmen den Überblick und die Kontrolle über den Softwarebereitstellungsprozess, was eine schnelle Reaktion auf Vorfälle ermöglicht und eine Kultur sicherer Programmierpraktiken fördert.

Best Practices

- [GAMESEC08-BP01 Wenden Sie Sicherheit in jeder Phase der CI/CD-Pipeline an](#)

GAMESEC08-BP01 Wenden Sie Sicherheit in jeder Phase der CI/CD-Pipeline an

Leitplanken wie Zugriffskontrollen, Aufgabentrennung und Prüfprotokolle bieten Schutz vor unbefugtem Zugriff oder böswilligen Aktivitäten.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Ihre Mitarbeiter, Prozesse und Technologien sollten die Pipeline ebenfalls sichern. Die Personen, die dem Code am nächsten stehen, müssen sichere Programmierpraktiken einrichten und sicherstellen, dass sie diese befolgen. Überprüfen Sie Ihre Prozesse kontinuierlich, um sicherzustellen, dass das Sicherheitsniveau in der gesamten Pipeline einheitlich ist. Implementieren Sie abschließend Technologien, um sicherzustellen, dass bewährte Verfahren und Prozesse nicht umgangen werden.

Kundenbeispiel

AnyCompany Games implementiert rollenbasierte Zugriffskontrollen, bei denen nur erfahrene Entwickler Änderungen an ihrem Anti-Cheat-Systemcode genehmigen können. Gleichzeitig müssen die Mitglieder des Sicherheitsteams für Komponenten, die Zahlungsdaten von Spielern verarbeiten, eine obligatorische Überprüfung des Codes vornehmen.

Ihre CI/CD Pipeline führt automatisch Überprüfungen der Bedrohungsmodelle durch und stellt so sicher, dass neue Funktionen wie ein Marktplatz für den Spielerhandel anhand zuvor identifizierter Angriffsvektoren wie Exploits zur Vervielfältigung von Objekten oder betrügerischen Transaktionsversuchen getestet werden.

Implementierungsschritte

- Erteilen Sie Benutzern Berechtigungen auf der Grundlage des Prinzips der geringsten Rechte.
- Wird verwendet AWS CloudTrail , um API-Aufrufe zu prüfen, die für die in der Pipeline verwendeten Dienste getätigt wurden.
- Verwenden Sie Pre-Commit-Hooks, um zu überprüfen, ob der Code den allgemeinen Praktiken und Unternehmensrichtlinien entspricht.

Automatisieren Sie die Sicherheit

GAMESEC09: Wie automatisieren Sie die Sicherheit in Ihrer CI/CD Pipeline?

Integrieren Sie Sicherheitsmaßnahmen in Ihre CI/CD Pipeline, um während des gesamten Entwicklungszyklus ein robustes Sicherheitsniveau aufrechtzuerhalten. Dieser Prozess bietet viele der gleichen Vorteile wie die Sicherheit Ihrer Pipeline. Eine sichere CI/CD Pipeline verringert die Wahrscheinlichkeit von Sicherheitsereignissen, die den Zeitplan für die Spieleentwicklung möglicherweise verzögern könnten.

Um Sicherheit in Ihrer CI/CD Pipeline zu gewährleisten, müssen Sie in jeder Phase des Entwicklungszyklus bewährte Sicherheitsmethoden und -tools implementieren. Wenn Sie Sicherheit in der Pipeline haben, können Sie auch den Zeitaufwand für Sicherheitsüberprüfungen reduzieren.

Die Automatisierung der Sicherheit innerhalb Ihrer CI/CD Pipeline ist besonders wichtig, um sicherzustellen, dass die Sicherheitskontrollen bei jeder Codeänderung konsistent implementiert und getestet werden. Durch die Implementierung der richtigen Tools und Automatisierung können sichere Spiele gewährleistet werden.

Best Practices

- [GAMESEC09-BP01 Integrieren Sie Tools und Automatisierung, um die durchschnittliche Zeit für Sicherheitsprüfungen zu reduzieren](#)

GAMESEC09-BP01 Integrieren Sie Tools und Automatisierung, um die durchschnittliche Zeit für Sicherheitsprüfungen zu reduzieren

Um Sicherheitslücken zu identifizieren, können Unternehmen eine Vielzahl verschiedener Tools und Dienste wie Static Application Security Testing (SAST) und Dynamic Application Security Testing (DAST) verwenden. SAST ist eine Möglichkeit, den Quellcode zu überprüfen und Sicherheitslücken zu ermitteln. DAST ist eine Blackbox-Methode zum Testen Ihres Codes, bei der Ihre Anwendungen getestet werden, ohne sich den Quellcode anzusehen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Ein weiteres Tool, das Unternehmen verwenden können, ist die Software Composition Analysis (SCA), mit der die Sicherheit Ihrer Drittanbieter- oder Open-Source-Abhängigkeiten bewertet wird. Für einen manuellere Ansatz können sichere Codeüberprüfungen während der gesamten Pipeline implementiert werden.

Kundenbeispiel

AnyCompany Games verwendet SAST-Tools, um potenzielle Sicherheitslücken während des Entwicklungsprozesses automatisch zu kennzeichnen. Sie verwenden auch DAST-Tools, um Bedrohungen gegen laufende Spielversionen zu simulieren und so zu überprüfen, ob die Sicherheitskontrollen wie vorgesehen funktionieren. Darüber hinaus integriert AnyCompany Games Tools zum Scannen von Abhängigkeiten in ihren Entwicklungsprozess, um bekannte Sicherheitslücken in Bibliotheken und Spiele-Engines von Drittanbietern automatisch zu identifizieren.

Implementierungsschritte

- Verwenden Sie Amazon CodeGuru als SAST-Tool.
- Verwenden Sie Open-Source-Tools wie OWASP Dependency Check,, oder. SonarQube OWASPZap

Ressourcen

- [Sicherheit für Entwickler](#)

Modellierung von Bedrohungen

GAMESEC10: Wie integrieren Sie die Bedrohungsmodellierung in den Lebenszyklus der Anwendungsentwicklung Ihres Unternehmens?

Bei der Bedrohungsmodellierung werden potenzielle Bedrohungen für Ihre Anwendung identifiziert und priorisiert und Lösungen ermittelt, mit denen sie gemindert werden können. Diese Praxis hat zunehmend an Bedeutung gewonnen, da sich Spiele zu komplexen, vernetzten Systemen entwickelt haben, die sensible Benutzerdaten und Echtgeldtransaktionen verarbeiten.

Integrieren Sie die Bedrohungsmodellierung als fortlaufende Maßnahme zur Unterstützung der Sicherheit des Spiels, und zwar nicht nur in der ersten Entwurfsphase, sondern auch im Zuge des weiteren Wachstums und der Weiterentwicklung des Spiels.

Best Practices

- [GAMESEC10-BP01 Ermitteln Sie, wann und wie Sie die Bedrohungsmodellierung während Ihres gesamten Anwendungsentwicklungszyklus durchführen müssen](#)

GAMESEC10-BP01 Ermitteln Sie, wann und wie Sie die Bedrohungsmodellierung während Ihres gesamten Anwendungsentwicklungszyklus durchführen müssen

Es gibt nicht die eine, beste Methode, um sich der Bedrohungsmodellierung zu nähern. Die Einzelheiten, wann und wie Sie dies tun können, hängen von den individuellen Anforderungen Ihres Spielestudios ab. Je nach Größe Ihres Studios können Sie beispielsweise Teammitglieder haben, die an einem oder mehreren Aspekten des Bedrohungsmodellierungsprozesses beteiligt sind.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Der [AWS Sicherheits-Blog](#) bietet einen Überblick über Überlegungen, die Sie bei der Entwicklung Ihrer Strategie zur Bedrohungsmodellierung berücksichtigen sollten, z. B.:

- Welche Ihrer Teammitglieder und Personas sollten an der Bedrohungsmodellierung beteiligt sein
- Wie bestimmt man die geeigneten Workflow-Tools, die verwendet werden sollen
- Wie lässt sich die Verantwortung für verschiedene Aspekte der Bedrohungsmodellierung ermitteln
- Wie können Sie Sicherheitskontrollen identifizieren und bewerten, die in Ihrem Workload-Design verwendet werden sollen

Kundenbeispiel

AnyCompany Spiele beginnen mit der Katalogisierung wertvoller Ressourcen wie Spielerdaten, Spielcode und Algorithmen, spielinterne Währungen, nutzergenerierte Inhalte und geistiges Eigentum wie unveröffentlichte Inhalte oder proprietäre Engines. Dabei werden verschiedene Arten potenzieller böswilliger Akteure berücksichtigt, z. B. Betrüger, die unfaire Vorteile suchen, böswillige Akteure, die

versuchen, persönliche oder finanzielle Daten zu stehlen, und böswillige Nutzer, die versuchen, das Gameplay zu stören.

Während des gesamten Entwicklungsprozesses verwendet AnyCompany Games Bedrohungsmodelle als Leitfaden für sichere Programmierpraktiken und beeinflusst Teststrategien, um sich auf Bereiche mit hohem Risiko zu konzentrieren. Vor der Veröffentlichung eines Spiels führen sie umfassende Überprüfungen der Bedrohungsmodelle durch, um die Bereitschaft zu beurteilen, ob die Spieler bereit sind, auf zu erwartende Belastungen und unbefugte Zugriffsversuche zu reagieren, und um Verfahren zur Reaktion auf Vorfälle vorzubereiten.

Implementierungsschritte

- Implementieren Sie in jeder Phase Ihrer CI/CD Pipeline Leitplanken.
- Verwenden Sie Automatisierung und Tools, um die Effizienz Ihrer Sicherheitsüberprüfungen für Anwendungen zu verbessern.
- Verwenden Sie die Bedrohungsmodellierung als Prozess zur Verbesserung der Sicherheit Ihrer Anwendungen.

Ressourcen

- [AWS Sicherheitsblog: Wie geht man mit der Bedrohungsmodellierung um](#)
- [NIST: Leitfaden zur datenzentrierten Modellierung von Systembedrohungen](#)
- [Bedrohungsmodellierung ist der richtige Weg für Bauherren — virtuelles AWS Skill Builder-Training zum Selbststudium](#)
- [Bedrohungsmodellierung für Bauherren — Workshop AWS](#)

Ressourcen

In den folgenden Ressourcen erfährst du mehr über unsere Best Practices im Bereich Sicherheit.

Zugehörige Dokumente:

- [Allgemeine Amazon Cognito-Szenarien](#)
- [Signiert verwenden URLs](#)
- [Verwenden Sie Channel-Flows, um vulgäre und sensible Inhalte aus Nachrichten in Amazon Chime SDK-Messaging zu entfernen](#)

- [Sicherheit bei Amazon GameLift](#)
- [Sichere Inhaltsbereitstellung mit Amazon CloudFront](#)
- [Leitfaden zur Reaktion auf Sicherheitsfragen](#)
- [AWS Bewährte Methoden für DDoS-Resiliency](#)
- [Schützen Sie Ihre AWS Cloud Umgebung vor Ransomware](#)

Ähnliche Partnerlösungen:

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

Verwandte Schulungsmaterialien:

- [Erste Schritte mit Amazon Cognito](#)
- [Schulung zur Sicherheit im Selbststudium](#)

Zuverlässigkeit

Zur Säule der Zuverlässigkeit gehört die Fähigkeit eines Systems, sich nach Infrastruktur- oder Dienstunterbrechungen zu erholen, Rechenressourcen dynamisch zu erwerben, um den Bedarf zu decken, und Störungen wie Fehlkonfigurationen oder vorübergehende Netzwerkprobleme zu minimieren.

Schwerpunktbereiche

- [Designprinzipien](#)
- [Grundlagen](#)
- [Workload-Architektur](#)
- [Änderungsmanagement](#)
- [Fehlerverwaltung](#)
- [Ressourcen](#)

Designprinzipien

Zusätzlich zu den Entwurfsprinzipien im Whitepaper AWS Well-Architected Framework sind die folgenden Entwurfsprinzipien aufgeführt, die die Zuverlässigkeit von Spiele-Workloads in der Cloud erhöhen können:

- Legen Sie die Ausgangswerte für die maximale Anzahl an Spielern und die Skalierbarkeit des Systems fest, die zur Erfüllung der Geschäftsprognosen erforderlich sind: Entwickeln Sie vor dem Start eines Spiels und während des Live-Spielbetriebs Schätzungen für die Anzahl der zu Spitzenzeiten zu erwartenden gleichzeitigen Spieler, um Zielziele für die Systemskalierbarkeit festzulegen, um diese Prognosen zu erfüllen. Auf diese Weise können Sie eine Grundlage für die Zuverlässigkeit Ihres Spiels schaffen. Definieren Sie Skalierungsrichtlinien, um Änderungen der Nachfrage automatisch zu berücksichtigen, ohne die Verfügbarkeit zu beeinträchtigen, indem Sie sicherstellen, dass Ihre Skalierungssysteme aktive Spielersitzungen ordnungsgemäß verwalten.
- Messen Sie Ihre Zuverlässigkeit und die Auswirkungen auf das Spielerlebnis: Definieren Sie wichtige Leistungsindikatoren (KPIs), die den Zustand Ihres Spiels widerspiegeln. Überwachen Sie die Auswirkungen von Änderungen an der Infrastruktur und den Spielfunktionen auf Ihre Zuverlässigkeit.

Grundlagen

Um Zuverlässigkeit zu erreichen, muss ein System über eine gut geplante Grundlage und Überwachung sowie Mechanismen zur Bewältigung von Bedarfs- oder Anforderungsänderungen verfügen. Das System sollte so konzipiert sein, dass es Fehler erkennt und sich automatisch repariert.

Für Games Lens gibt es keine spezifischen Grundlagen und bewährte Verfahren. Im Whitepaper Well-Architected Framework finden Sie bewährte Methoden in Bezug auf [Grundlagen](#) der Zuverlässigkeit, die für Spiele-Workloads gelten.

Workload-Architektur

GAMEREL01: Nutzt Ihre Spielarchitektur die Ausfallsicherheit der Cloud?

AWS Die Infrastruktur basiert auf Regionen und Availability Zones. AWS-Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Diese Konstrukte können verwendet werden, um Workloads so zu gestalten, dass Zuverlässigkeitsziele im Mittelpunkt stehen.

Best Practices

- [GAMEREL01-BP01 Verteilen Sie die Spieleinfrastruktur auf mehrere Availability Zones und Regionen, um die Ausfallsicherheit zu verbessern](#)

GAMEREL01-BP01 Verteilen Sie die Spieleinfrastruktur auf mehrere Availability Zones und Regionen, um die Ausfallsicherheit zu verbessern

Um die Auswirkungen lokaler Infrastrukturbeeinträchtigungen auf Ihre Spieler so gering wie möglich zu halten, sollten Sie Ihre Infrastruktur gleichmäßig auf genügend unabhängige Standorte verteilen, um unerwartete Beeinträchtigungen aushalten zu können und gleichzeitig über genügend Kapazität zu verfügen, um den Bedürfnissen Ihrer Spieler gerecht zu werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Bei der Bereitstellung Ihrer Spieleinfrastruktur wird empfohlen, Ihre Kapazität gleichmäßig auf mehrere Availability Zones in einer Region zu verteilen, damit Sie Störungen in einer oder mehreren Availability Zones überstehen können, ohne das Spielerlebnis zu beeinträchtigen. Backend-Dienste für Spiele wie Webanwendungen sollten über mehrere Availability Zones verteilt werden oder mithilfe von verwalteten Diensten wie AWS Lambda Amazon API Gateway erstellt werden, die von Haus aus regionale Hochverfügbarkeit bieten. In ähnlicher Weise sollten Komponenten, die den Status beibehalten, wie Caches, Datenbanken, Nachrichtenwarteschlangen und Speicherlösungen, so konzipiert sein, dass sie eine dauerhafte Persistenz von Daten über mehrere Availability Zones hinweg gewährleisten, was in Services wie Amazon S3, DynamoDB und Amazon SQS standardmäßig vorgesehen ist und in anderen Diensten konfiguriert werden kann.

Wenn Sie Ihre Gameserver-Hosting-Architektur auf Ausfallsicherheit auslegen, sollten Sie Ihre Spieleserverflotten innerhalb der Availability Zones einheitlich bereitstellen, AWS-Region um Ihren Zugriff auf die verfügbare Rechenkapazität in der Region zu maximieren und das Ausmaß der Auswirkungen von Beeinträchtigungen der Availability Zone zu verringern. Sie können [Amazon EC2 Auto Scaling](#) beispielsweise so konfigurieren, dass es die Availability Zones verwendet. Wenn eine EC2 Instance fehlerhaft wird, kann EC2 Auto Scaling die Instance ersetzen und Instances in anderen Availability Zones starten, falls eine oder mehrere der Availability Zones nicht mehr verfügbar sind.

Stellen Sie für kritische Infrastrukturen wie Authentifizierung eine Mindestanzahl funktionsfähiger Instances bereit, die über mehrere Availability Zones laufen, und verwenden Sie Auto Scaling, um Lasterhöhungen oder Fehlertoleranz zu bewältigen, falls eine der Availability Zones beeinträchtigt wird.

Stellen Sie Ihre Spiele-Infrastruktur in mehreren Regionen bereit, um die Verfügbarkeit zu maximieren. Regionalübergreifende Disaster-Recovery-Funktionen wie globale Aurora-Datenbanken und eine redundante Infrastruktur, die mit einer einfachen DNS-Änderung in einer sekundären Region aktiv werden kann, können die Servicekontinuität gewährleisten, falls die primäre Region beeinträchtigt wird. Wir empfehlen dies zwar, damit eure Backend-Dienste für Spiele eine hohe Verfügbarkeit erreichen, aber diese Empfehlung ist besonders wichtig für eure Spieleserver.

In einem Multiplayer-Spiel beispielsweise wird Ihre Infrastrukturkapazität für Spieleserver wahrscheinlich die Kapazitätsanforderungen für Ihre anderen Dienste übersteigen, da Spieleserver dazu verwendet werden, Spielsitzungen für Spieler zu hosten. Viele Spiele entscheiden sich dafür, Spieler in logische Spielregionen (wie den Westen und Osten der USA) aufzuteilen. Um das Spielerlebnis zu vereinfachen und die Nutzung der globalen Infrastruktur für das Hosten von

Spielen zu vereinfachen, sollten Sie erwägen, den Namen Ihrer spielerseitigen Spielregionen von der zugrunde liegenden Cloud-Anbieter-Region oder dem Rechenzentrumsstandort zu trennen, der die Spieleserver physisch hostet, zusammen mit anderer Infrastruktur wie Local Zones oder Ihren eigenen Rechenzentren, die Spieleserver-Instanzen hosten, die diese Spielerspielregion unterstützen.

Setzen Sie bei der Gestaltung Ihres Spielerzuweisungsdienstes auf eine Architektur mit mehreren Regionen und separater Softwarebereitstellung in allen Regionen. Entkoppeln Sie den Einsatz Ihres Matchmaking-Dienstes von den Flotten, die Ihre Spieleserver-Instanzen hosten, sodass Sie Spieler zu einem Spieleserver in Regionen weiterleiten können, unabhängig davon, welche regionale Einrichtung Ihres Matchmaking-Dienstes die Spielerzuweisungsanfrage bearbeitet hat.

Entwerfe die Logik in deiner Matchmaking-Implementierung so, dass du die Spieleserver-Regionen bevorzugst, die deine Latenz und andere Regeln einhalten, mit der Möglichkeit, Spieler in andere Regionen weiterzuleiten, wenn deine Flotten wenig Kapazität haben oder es zu anderen regionalen Infrastrukturunterbrechungen kommt.

Implementierungsschritte

- Verteilen Sie die Spielinfrastruktur einheitlich auf mehrere Availability Zones, um eine hohe Verfügbarkeit und Ausfallsicherheit zu gewährleisten.
- Stellen Sie Backend-Services und Stateful-Komponenten für Spiele mithilfe von Managed Services wie AWS Lambda Amazon S3, DynamoDB und SQS bereit oder konfigurieren Sie Load Balancing und Haltbarkeit für benutzerdefinierte Lösungen.
- Implementieren Sie regionsübergreifende Bereitstellungen für wichtige Spieledienste und Server mithilfe von Disaster-Recovery-Lösungen wie globalen Aurora-Datenbanken und logischen Regionen mit Spielerkontakt, die von den zugrunde liegenden physischen Standorten entkoppelt sind.

Ressourcen

- [Statische Stabilität](#)
- [Bewährte Methoden für Warteschlangen Amazon GameLift Amazon-Spielesitzungen](#)
- [GameLift Amazon-Flotten mit mehreren Regionen](#)
- [Aurora Global Database](#) für Disaster Recovery

Änderungsmanagement

GAMEREL02: Wie skalieren Sie Ihre Stateful-Spiele, um Änderungen der Nachfrage Rechnung zu tragen?

Da die Nachfrage Ihrer Spieler im Laufe der Zeit schwankt, sollte Ihre Spieleinfrastruktur in der Lage sein, sich an diese sich ändernden Anforderungen anzupassen. Es ist zwar schwierig, die Beliebtheit eines Spiels im Voraus vorherzusagen, aber entwerfen Sie einen Architekturansatz, der das Hinzufügen und Entfernen von Infrastrukturkapazitäten ermöglicht, um Schwankungen in der Spielerpopulation Rechnung zu tragen.

Best Practices

- [GAMEREL02-BP01 Implementieren Sie eine Skalierungsstrategie, die den Status der aktiven Spielsitzungen berücksichtigt](#)
- [GAMEREL02-BP02 Support die Verwendung mehrerer EC2 Instanztypen für dein Spiel](#)

GAMEREL02-BP01 Implementieren Sie eine Skalierungsstrategie, die den Status der aktiven Spielsitzungen berücksichtigt

Implementieren Sie eine Lösung für die automatische Skalierung Ihrer Spieleinfrastruktur. Dabei wird der statusbehaftete Charakter Ihrer aktiv verbundenen Spielsitzungen berücksichtigt und Skalierungsaktivitäten problemlos bewältigt, ohne das Gameplay zu stören.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Einer der Vorteile der Entwicklung eines Spiels in der Cloud ist die Elastizität, die durch die automatische Skalierung der Serverinfrastruktur nach Bedarf erreicht werden kann, um den Bedarf zu decken. Zustandslose oder asynchrone Spiele und Backend-Services können zwar mithilfe von [Amazon EC2 Auto Scaling-Richtlinien, EKS-Autoscaling](#) oder ähnlichen Techniken, die typischerweise für skalierbare Webanwendungen verwendet werden, dynamisch skaliert werden. Spieleentwickler benötigen jedoch in der Regel einen individuelleren Ansatz für die Skalierung von statusbehafteten oder synchronen Spielen, um Störungen bei aktiven Spielsitzungen zu verhindern.

Implementierungsschritte

- Generieren Sie für statusbehaftete Spiele benutzerdefinierte Messwerte, mit denen Sie den Status Ihrer Spielsitzungen und die verfügbare Spieleserverkapazität überwachen können. Diese können Amazon CloudWatch als benutzerdefinierte Messwerte gemeldet werden. Trainingsfunktionen mit Anwendungsüberwachung, z. B. CloudWatch Synthetics, bei denen das Spiel auf Beeinträchtigungen von Funktionen überprüft wird, die durch einfaches Hoch- und Herunterfahren des Zustands möglicherweise nicht erkannt werden.
- Implementieren Sie mithilfe benutzerdefinierter Metriken Software zur Gameserver-Skalierung, z. B. als serverlose Anwendung, die AWS Lambda Funktionen verwendet AWS Fargate, oder um die Flotte der dedizierten Spieleserver-Instanzen zu verwalten, indem Sie das AWS SDK verwenden, um API-Aufrufe zu tätigen, um die Mindest-, Maximal- und gewünschten Kapazitätseinstellungen für die EC2 [Auto Scaling Scaling-Gruppen](#) zu aktualisieren, die Ihren Spielserver-Build hosten.
- Verwenden Sie Amazon GameLift, um Ihre Spieleserver zu hosten, und nutzen Sie die [auto Skalierungsfunktionen des out-of-the-box Spiele_servers](#), um diesen Skalierungsprozess für Sie zu verwalten.

Die automatischen Skalierungsfunktionen von Amazon GameLift erkennen aktive Spielsitzungen und können so konfiguriert werden, dass sie die Beendigung oder Skalierung von Spielserver-Instances blockieren, die aktiv Spieler hosten. Weitere Informationen finden Sie unter [Überwachen von GameLift Amazon-Servern mit Amazon CloudWatch](#).

GAMEREL02-BP02 Support die Verwendung mehrerer EC2 Instanztypen für dein Spiel

Verwenden Sie in Ihrer Hosting-Strategie mehrere EC2 Instance-Typen, wenn Sie Ihr Spiel mithilfe von EC2 Instances hosten oder wenn Sie Container verwenden AWS-Konto, die auf Ihren Instanzen gehostet werden. Durch die Verwendung mehrerer Instance-Typen kannst du die Anzahl der Rechenoptionen erhöhen, die verwendet werden können, wenn dein Spiel skaliert wird, um mehr Server hinzuzufügen, um das Spielerwachstum zu unterstützen. Das erhöht die Zuverlässigkeit für den Fall, dass dein bevorzugter Instance-Typ nicht verfügbar ist.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Wenn du dein Spiel mithilfe von Spot-Instances hostest, verwende mehrere Instance-Typen, da die Verfügbarkeit von Spot-Instances je nach Kundennachfrage schwankt.

Testen Sie Ihr Spiel auf mehreren Instance-Typen, um Ihre Kosten- und Leistungsanforderungen zu erfüllen, und legen Sie eine priorisierte Rangfolge der Instance-Typen fest. Amazon EC2 Auto Scaling unterstützt die Verwendung mehrerer Instance-Typen und -Größen sowie die [Zuweisung von Gewichtungen zu jedem Instance-Typ](#) in Ihrer Konfiguration, sodass Sie eine priorisierte Rangfolge der Rechenoptionen implementieren können.

Wenn Sie Ihr Spiel mit Amazon GameLift Managed Hosting hosten, entscheiden Sie, welche Art von Instanzen Ihr Spiel benötigt und wie die Spielserverprozesse darauf ausgeführt werden sollen (mithilfe einer Laufzeitkonfiguration). Bei der Auswahl der Ressourcen für eine Flotte sollten Sie mehrere Faktoren berücksichtigen, darunter das Betriebssystem des Spiels, den Instance-Typ (die Computerhardware) und ob On-Demand-Instances, Spot-Instances oder beides verwendet werden sollen. Die Hosting-Kosten bei Amazon hängen GameLift in erster Linie von der Art der Instances ab, die Sie verwenden. Weitere Informationen finden [Sie unter Wählen Sie Rechenressourcen für eine verwaltete Flotte](#).

Implementierungsschritte

- Verwenden Sie mehrere EC2 Instance-Typen, um die Zuverlässigkeit und Skalierungsoptionen zu verbessern, wenn Sie Ihr Spiel auf EC2 oder in Containern hosten.
- Konfigurieren Sie Amazon EC2 Auto Scaling oder GameLift Flotten mit priorisierten Instance-Typen und Gewichtungen, um Kosten und Leistung zu optimieren.
- Testen Sie Ihr Spiel auf verschiedenen Instance-Typen, um sicherzustellen, dass die Leistung den Anforderungen entspricht, und passen Sie Ihre Hosting-Strategie entsprechend an.

Fehlerverwaltung

GAMEREL03: Wie hält man den Spielstatus bei Infrastrukturunterbrechungen aufrecht?

Da Ihre Spielinfrastruktur im Laufe der Zeit verschiedenen betrieblichen Ereignissen ausgesetzt ist, sollte die Architektur Ihres Spiels darauf ausgelegt sein, die Kontinuität der Spielerlebnisse und den Spielstatus bei Infrastrukturreignissen aufrechtzuerhalten. Implementieren Sie zur Bewältigung dieser Ereignisse Mechanismen zur Überwachung, ordnungsgemäßen Herunterfahren und Statuspersistenz, um sicherzustellen, dass Ihre Spieler ein reibungsloses Spielerlebnis haben.

Best Practices

- [GAMEREL03-BP01 Überwachen Sie Störungen des Spieleservers und verwenden Sie die Daten, um die Hosting-Architektur zu verbessern und so die Zuverlässigkeitsziele zu erreichen](#)
- [GAMEREL03-BP02 Implementieren Sie eine lose Kopplung von Spielfunktionen, um Fehler mit minimaler Beeinträchtigung des Spielerlebnisses zu beheben](#)
- [GAMEREL03-BP03 Überwachen Sie Infrastrukturereignisse im Laufe der Zeit, um die Auswirkungen auf das Spielerverhalten zu messen](#)

GAMEREL03-BP01 Überwachen Sie Störungen des Spieleservers und verwenden Sie die Daten, um die Hosting-Architektur zu verbessern und so die Zuverlässigkeitsziele zu erreichen

Überwachen Sie die Spielserver-Metriken und die Auswirkungen von Ausfällen oder Leistungseinbußen, wie z. B. einer erhöhten Latenz unter Last, auf das Spielerverhalten im Laufe der Zeit, sodass Sie Ihre Hosting-Strategie für Spieleserver an die Zuverlässigkeitsanforderungen Ihres Spiels anpassen können. Die Spielserverinfrastruktur, die herabgesetzt werden soll, sollte sofort außer Betrieb genommen werden, wenn sie sich negativ auf die Spieler auswirkt, oder proaktiv ersetzt werden, wenn keine aktiven Spielersitzungen auf dem Server gehostet werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

In Szenarien, in denen Spiele als REST gehostet werden APIs, kann die Systemzuverlässigkeit wie bei herkömmlichen Webanwendungsarchitekturen verwaltet werden, bei denen der Datenverkehr verteilt auf mehrere Server verteilt werden kann, um das Risiko von Serverausfällen zu minimieren.

Bei synchronem Echtzeit-Gameplay wird eine Spielsitzung normalerweise auf einem Spieleserverprozess gehostet, der auf einer virtuellen Maschine oder einer Spielserverinstanz ausgeführt wird, da der Gameplay-Status performant aufrechterhalten und auf verbundene Spieleclients repliziert werden muss. Diese Implementierung bedeutet, dass das Spielerlebnis eng mit der Leistung und Zuverlässigkeit des Gameserverprozesses verknüpft ist, der die Spielsitzung hostet. Diese Art von Architektur macht die Verwaltung der Zuverlässigkeit von Spieleservern komplexer als bei herkömmlichen Ansätzen.

[Um die Auswirkungen eines Spielerverausfalls zu minimieren, konfigurieren Sie Ihr Spiel so, dass der Spielstatus eines Spielers kontinuierlich asynchron in einem hochverfügbaren Cache oder einer](#)

[Datenbank wie Amazon ElastiCache \(Redis OSS\) oder Amazon MemoryDB aktualisiert wird.](#) Wenn ein Serverausfall auftritt, kann der zuletzt gespeicherte Spielstatus des Spielers aus dem externen Datenspeicher abgerufen werden, und seine Sitzung kann auf einer neuen Spielservers-Instanz wiederhergestellt werden.

Dieser Ansatz verursacht jedoch zusätzliche Kosten und Komplexität bei der Verwaltung dieses externen Zustands und ist möglicherweise nicht für schnelle oder wettbewerbsorientierte Spiele geeignet, bei denen die Statusänderungen so häufig und in einem so großen Umfang auftreten, dass selbst die Einführung eines leistungsfähigen In-Memory-Cache-Datenspeichers zu einer Replikationsverzögerung führen würde, die zu groß ist, um eine Sitzung wiederherzustellen. Bei Spielen dieser Art besteht der optimale Ansatz darin, den Verlust des Servers zu akzeptieren und den Spieler zurück in eine Spielelobby zu schicken, um eine andere Sitzung zu finden, oder Sie können ihn automatisch in eine andere Spielsitzung umleiten.

Erfassen Sie möglichst viele nützliche Protokolldaten darüber, was die Serverunterbrechung verursacht hat, damit Sie das Problem später untersuchen können. Amazon GameLift bietet Anleitungen zum [Debuggen von Flottenproblemen](#) und bietet die Möglichkeit, [remote auf GameLift Amazon-Flotteninstanzen zuzugreifen](#).

Implementierungsschritte

- Überwachen Sie die Spielservers-Metriken auf Leistungseinbußen und entfernen oder ersetzen Sie heruntergekommene Server nach Bedarf, um die Zuverlässigkeit aufrechtzuerhalten.
- Verwenden Sie Amazon ElastiCache oder MemoryDB für asynchrone Spielstatus-Updates, um die Sitzungswiederherstellung nach Serverausfällen zu ermöglichen, sofern dies möglich ist.
- Erfassen Sie detaillierte Protokolldaten zu Serverunterbrechungen zur Untersuchung und zum Debuggen und nutzen Sie Tools wie Amazon GameLift für die Flottenüberwachung und den Fernzugriff.

GAMEREL03-BP02 Implementieren Sie eine lose Kopplung von Spielfunktionen, um Fehler mit minimaler Beeinträchtigung des Spielerlebnisses zu beheben

Das Entkoppeln von Komponenten bezieht sich auf das Konzept, Serverkomponenten so zu entwerfen, dass sie so unabhängig wie möglich arbeiten können. Einige Aspekte des Spielens lassen sich nur schwer entkoppeln, da die Daten so aktuell wie möglich sein müssen, um den Spielern ein gutes Spielerlebnis zu bieten. Viele Komponenten und Spielaufgaben können jedoch entkoppelt

werden. Zum Beispiel sind Bestenlisten und Statistikdienste für das Spielerlebnis nicht entscheidend, und die Lese- und Schreibvorgänge in diesen Diensten können asynchron vom Spiel aus ausgeführt werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Implementieren Sie eine schrittweise Degradation für Funktionen in Ihrem Spiel, die automatisch oder von einem Administrator deaktiviert werden können, wenn Probleme erkannt werden, und konfigurieren Sie Upstream-Dienste, die von der Funktion abhängen, um den Ausfall ordnungsgemäß behandeln zu können. Wenn beispielsweise bestimmte Spielerdaten in deinem Spielclient nicht richtig geladen werden, solltest du dir überlegen, ob diese Daten für das Spielerlebnis entscheidend sind. Falls nicht, konfiguriere den Spielclient so, dass er diesen Fehler behutsam behandelt, ohne das Spielerlebnis zu beeinträchtigen, und wähle, diese Daten später erneut abzurufen, wenn der Spieler den Bildschirm erneut aufruft.

Nutze Logik wie Timeouts, Wiederholungen und Backoff, um Fehler und Ausfälle zu behandeln. Timeouts verhindern, dass Systeme für unangemessen lange Zeiträume hängen bleiben. Wiederholungen können für eine hohe Verfügbarkeit vorübergehender und zufälliger Fehler sorgen.

Definieren Sie unkritische Komponenten, die lose an kritische Komponenten gekoppelt werden können. Durch die lose Kopplung können Systeme widerstandsfähiger sein, da sich der Ausfall einer Komponente nicht auf andere überträgt. Wenn für Spielfunktionen keine statusbehafteten Verbindungen zu Ihren Spieleservern oder Ihrem Backend erforderlich sind, sollten Sie statusfreie Protokolle implementieren, um dynamisch zu skalieren und vorübergehende Ausfälle zu beheben. Entwickeln Sie Ihre unkritischen Komponenten so, dass sie mithilfe einer API lose mit zustandslosen Protokollen verknüpft werden können. HTTP/JSON Implementieren Sie Netzwerkaufrufe vom Spielclient so, dass sie asynchron und nicht blockierend erfolgen, um die Auswirkungen langsamer Spielfunktionen oder anderer abhängiger Dienste auf die Spieler zu minimieren.

Um die Ausfallsicherheit durch lose Kopplung weiter zu verbessern, verwenden Sie einen Messaging-Dienst wie Warteschlangen, Streaming oder ein themenbasiertes System zwischen Komponenten, die asynchron verarbeitet werden können. Dieses Modell eignet sich für Interaktionen, die keine sofortige Antwort erfordern oder bei denen eine Bestätigung, dass eine Anfrage registriert wurde, ausreichend ist. Diese Lösung beinhaltet eine Komponente, die Ereignisse generiert, und eine andere, die sie verarbeitet. Die beiden Komponenten werden nicht durch direkte point-to-point Interaktion integriert, sondern über ein Zwischenprodukt wie eine dauerhafte Speicher- oder

Warteschlangenschicht. Dies trägt auch dazu bei, die Zuverlässigkeit des Systems zu verbessern, indem Nachrichten gespeichert werden, wenn die Verarbeitung fehlschlägt.

Suchen Sie nach einem geeigneten Nachrichtenmechanismus und wählen Sie ihn aus, da verschiedene Nachrichtendienste unterschiedliche Merkmale aufweisen, z. B. Bestell- und Zustellungsmechanismen. Entwerfen Sie Operationen so, dass sie idempotent sind, sodass das gewählte Nachrichtensystem Nachrichten mindestens einmal zustellt. Stellen Sie sich als Beispiel einen typischen Anwendungsfall für Spiele vor, bei dem Ihr Spiel die Spielzeit, Statistiken oder andere relevante Daten von Spielern verfolgen muss, was zu einem Anwendungsfall mit hohem Schreibdurchsatz in Zeiten hoher Parallelität der Spieler führen kann.

Um eine zuverlässige Architektur zu implementieren, sollten Sie sich überlegen, ob für den Anwendungsfall eine vom Spieler wahrgenommene read-after-write Konsistenz erforderlich ist. In der Regel eignen sich solche Szenarien für die asynchrone Verarbeitung und können durch die Implementierung eines Schreibwarteschlangenmusters erreicht werden, bei dem die Anfragen in eine skalierbare und dauerhafte Nachrichtenwarteschlange wie Amazon SQS aufgenommen und mithilfe eines Verbraucherdienstes, z. B. einer Lambda-Funktion, stapelweise in Ihre Backend-Datenbank eingefügt werden können. Dieser Ansatz ist zuverlässiger als die synchrone Kommunikation zwischen mehreren verteilten Komponenten, einschließlich dem Spielclient des Spielers, Ihren Backend-Web- und Anwendungsservern und Ihrem internen Datenbanksystem. Es reduziert auch die Kosten, da die Backend-Datenbank nicht skaliert werden muss, um den maximalen Schreibdurchsatz zu bewältigen, da die Benutzerverarbeitung aus der Schreibwarteschlange genutzt werden kann, um diese Aufnahmezeit nach Bedarf zu verlangsamen.

Implementierungsschritte

- Entkoppeln Sie unkritische Komponenten wie Bestenlisten und Statistikdienste von wichtigen Spielfunktionen, um asynchrone Abläufe zu ermöglichen und die Stabilität zu erhöhen.
- Implementieren Sie eine ordentliche Degradation für unkritische Funktionen mit Logik für Timeouts, Wiederholungen und Backoff und stellen Sie sicher, dass der Spielclient Ausfälle verarbeitet, ohne das Spielerlebnis zu beeinträchtigen.
- Verwenden Sie Messaging-Systeme wie Amazon SQS für die asynchrone Kommunikation zwischen Komponenten und ermöglichen Sie so eine skalierbare, dauerhafte und zuverlässige Verarbeitung von Anwendungsfällen mit hohem Durchsatz.

Ressourcen

- [Erstellen Sie mithilfe der Microservice-Architektur hochgradig skalierbare und zuverlässige Workloads](#)
- [Integration von Microservices mithilfe serverloser Dienste AWS](#)
- [Grundlegendes zu asynchronem Messaging für Microservices](#)
- [Einführung in skalierbare Spieleentwicklungsmuster auf AWS](#)
- Implementierung von [Graceful Degradation](#)

GAMEREL03-BP03 Überwachen Sie Infrastrukturereignisse im Laufe der Zeit, um die Auswirkungen auf das Spielerverhalten zu messen

Überwachen Sie den Prozess Ihres Spieleservers, die Messwerte für Ihre Spieleserverinstanz und das Spielerlebnis, um die Hauptursache von Problemen zu ermitteln. Zusätzlich zur Überwachung von CPU und Arbeitsspeicher können Sie auch die Überwachung von Netzwerkmetriken einrichten, die sich auf Netzwerkbeschränkungen von EC2 Instances beziehen, um Sie auf Probleme wie Bandbreitenüberschreitung oder andere Probleme auf Netzwerkebene aufmerksam zu machen packets-per-second, die darauf hindeuten können, dass Ihre Serverressourcen nicht ausreichend bereitgestellt sind.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Verwenden Sie CloudWatch Synthetics, um die Funktionalität kritischer Pfadanwendungen für das Spielerlebnis zu überprüfen, z. B. wenn Sie sich nicht anmelden können oder andere Probleme, die sich auf den Dienst auswirken. Erwägen Sie für Spieleserver, die über Amazon gehostet werden GameLift, die Überwachung von [Metriken](#) wie:

- GameServerInterruptions und InstanceInterruptions, was Ihnen helfen kann zu verstehen, wie sich Einschränkungen bei der Verfügbarkeit von Spot-Instances auf Ihre mit Spot bereitgestellten Spieleserver auswirken.
- ServerProcessAbnormalTerminations, das verwendet werden kann, um ungewöhnliche Abbrüche in Ihren Spieleserverprozessen zu erkennen.

Es wird empfohlen, historische Messdaten zur Zuverlässigkeit Ihres Spieleservers aufzubewahren. Verwenden Sie diese historischen Daten für Berichtszwecke und verknüpfen Sie sie mit anderen

Datensätzen, um potenzielle Trends aufzudecken und die Auswirkungen auf das Spielerverhalten im Laufe der Zeit zu bewerten, die möglicherweise auf Probleme mit dem Spielserver zurückzuführen sind.

Amazon CloudWatch speichert Metriken nicht unbegrenzt, und die [Speicherauflösung von Metriken](#) wird im Laufe der Zeit erhöht. Erwägen Sie daher, diese Metriken in einen kostengünstigen Langzeitspeicher wie Amazon S3 zu exportieren. Sie können [CloudWatchMetric Streams](#) so konfigurieren, dass Ihre Metriken automatisch aus [CloudWatchRegionen](#) in Ihren eigenen S3-Bucket übertragen werden, wo sie langfristig in einer Speicherebene wie S3 Intelligent-Tiering gespeichert und schließlich mit Amazon Glacier archiviert werden können. Wenn Sie Ihre Metriken in Amazon S3 platzieren, können sie sofort mit anderen Datensätzen in Ihrem Data Lake für interaktive Abfragen mit [Amazon Athena](#) verknüpft werden.

Implementierungsschritte

- Überwachen Sie Gameserver-, Instance- und Netzwerkmetriken, einschließlich Bandbreite und packet-per-second Limits, mithilfe von Amazon CloudWatch und CloudWatch Synthetics für Funktionsprüfungen kritischer Pfade.
- Verfolgen Sie GameLift spezifische Messwerte wie GameServerInterruptions und ServerProcessAbnormalTerminationsum deren Auswirkungen auf die Verfügbarkeit von Spot-Instances zu bewerten und ungewöhnliche Serverabbrüche zu erkennen.
- Exportieren Sie CloudWatch Metriken zur Langzeitspeicherung nach Amazon S3, verwenden Sie kostengünstige Stufen wie S3 Intelligent-Tiering oder Glacier und analysieren Sie Trends mit Tools wie Amazon Athena.

Ressourcen

- [Netzwerkleistungskennzahlen auf EC2 Amazon-Instanzebene ermöglichen neue Erkenntnisse](#)
- [CloudWatchMetric Streams — Senden Sie AWS Metriken in Echtzeit an Partner und an Ihre Apps](#)

Ressourcen

In den folgenden Ressourcen erfahren Sie mehr über unsere Best Practices im Bereich Zuverlässigkeit.

Zugehörige Dokumente:

- [Wir praktizieren Continuous Integration und Continuous Delivery am AWS](#)

- [Automatische Skalierung asynchroner Jobwarteschlangen](#)
- [Entwerfen Sie Ihre Architektur WorkloadService](#)
- [Timeouts, Wiederholungsversuche und Backoff mit Jitter](#)
- [Well-Architected Framework — Säule der Zuverlässigkeit](#)
- [Architektur für zuverlässige Skalierbarkeit](#)
- [Die Amazon Builder-Bibliothek](#)
- [Umfangreiches Echtzeit-Messaging für Multiplayer-Spiele](#)
- [Einführung in skalierbare Spielentwicklungsmuster auf AWS](#)
- [Containerisierte Microservices ausführen auf AWS](#)
- [Hosting von Webanwendungen in der Cloud](#)
- [Aufbau einer skalierbaren und sicheren Multi-VPC-Netzwerkinfrastruktur](#)

Zugehörige Videos:

- [re:Invent 2020: Ubisoft: Entwicklung eines plattformübergreifenden Multiplayer-Spiels auf AWS](#)
- [re:Invent 2018: Supercell — Skalierung von Handyspielen](#)
- [re:Invent 2019: Wie CAPCOM unterhaltsame Spiele mit Containern, Daten und ML entwickelt](#)
- [re:Invent 2018: Globalisierung der Spielerkonten bei Riot Games bei gleichbleibender Verfügbarkeit](#)
- [re:Invent 2020: GameLoft — Ein tiefer Einblick in die Data Lake-Migration ohne Ausfallzeiten](#)

Zugehörige Schulungen:

- [Amazon GameLiftFleet IQ für Spieleserver verwenden](#)
- [Gameserver-Hosting bei Amazon EC2](#)

Leistungseffizienz

Der Schwerpunkt der Leistungseffizienz liegt auf der effizienten Nutzung von Rechenressourcen zur Erfüllung der Anforderungen und der Aufrechterhaltung dieser Effizienz, wenn sich die Nachfrage ändert und sich die Technologien weiterentwickeln.

Wählen Sie bei der Auswahl einer Hochleistungsarchitektur einen datengesteuerten Ansatz. Sammeln Sie umfassende Daten zur Architektur, vom Entwurf auf hoher Ebene bis hin zur Auswahl und Konfiguration von Ressourcentypen. Berücksichtigen Sie Ihre architektonischen Entscheidungen auf zyklischer Basis, um die Vorteile des sich ständig weiterentwickelnden Angebots an Services und Lösungen zu nutzen. Kennzahlen helfen dabei, Abweichungen von der erwarteten Leistung zu verstehen, sodass Sie Maßnahmen ergreifen können. Ein datengestützter Ansatz hilft Ihnen dabei, Kompromisse mit Ihrer Architektur einzugehen, um die Leistung zu verbessern, die Kosten zu senken oder das Entwicklererlebnis zu verbessern.

Schwerpunktbereiche

- [Designprinzipien](#)
- [Auswahl der Architektur](#)
- [Auswahl der Region](#)
- [Iterative Entwicklung](#)
- [Computer und Hardware](#)
- [Auswahl berechnen](#)
- [Datenverwaltung](#)
- [Netzwerk und Bereitstellung von Inhalten](#)
- [Prozess und Kultur](#)
- [Ressourcen](#)

Designprinzipien

Zusätzlich zu den Designprinzipien im Whitepaper AWS Well-Architected Framework können Sie mit den folgenden Designprinzipien die Leistungseffizienz Ihrer Spiele steigern:

- Messen Sie die Spielleistung end-to-end anhand von: Es ist wichtig, die Leistung so zu messen, wie sie aus der Sicht Ihrer Spieler wahrgenommen wird. Das bedeutet, dass du die Leistung des Spielclients, deiner Spielinfrastruktur und der Internetverbindung, die deine Spieler mit der

Infrastruktur verbindet, messen solltest. Das hilft dir zu verstehen, wo du die Leistung deiner Architektur verbessern kannst.

- Optimieren Sie Ihre Architektur, um die Kennzahlen zu verbessern, die das tatsächliche Spielerlebnis widerspiegeln: Wenn Sie Ihre Architektur im Laufe der Zeit anpassen und weiterentwickeln, sollten Sie darüber nachdenken, wie sich diese Verbesserungen und Änderungen auf das Spielerlebnis auswirken werden. Die Workloads in Spielen sollten in der Lage sein, den Auswirkungen von Ausfällen standzuhalten und diese so gering wie möglich zu halten, um weitverbreitete Spielunterbrechungen zu verhindern. Spielfunktionen und Systeme, die nicht entscheidend voneinander abhängig sind, sollten entkoppelt werden, um den Explosionsradius von Ausfällen zu verringern und Probleme zu isolieren, die sich auf die Spieler auswirken.
- Verwenden Sie Technologien, die den Spielbetrieb vereinfachen und die Entwicklungsgeschwindigkeit erhöhen: Priorisieren Sie die Einführung von Technologien, die die Effizienz der Entwickler verbessern können. Der betriebliche Aufwand in der Vorproduktionsphase der Entwicklung kann von der Verbesserung des Gameplays ablenken. Durch die Nutzung von Managed Services von AWS AWS unseren Partnern kann der technische Aufwand reduziert werden, sodass sich Spieleentwickler auf den eigentlichen Spielablauf und das Spielerlebnis konzentrieren können. Architektur- und Leistungsanforderungen können sich während des gesamten Lebenszyklus der Spieleentwicklung ändern und weiterentwickeln, und technische Kompromisse sollten in jeder Phase berücksichtigt werden.
- Die Infrastruktur sollte so konzipiert sein, dass sie auch in Spitzenzeiten gleichzeitig spielt, und dynamisch nach Bedarf skaliert werden kann: Die Infrastruktur sollte so konzipiert sein, dass sie der Nachfrage der Spieler gerecht wird. Metriken wie die Parallelität der Spielsitzungen und die Anzahl der Anmeldungen können verwendet werden, um präventiv zu skalieren, bevor die Systeme überlastet werden. Messwerte zur reaktiven Systemauslastung, wie z. B. CPU- und Speicherverbrauch, können zur Skalierung verwendet werden, wenn Systeme überlastet sind. Durch die dynamische Skalierung deiner Infrastruktur kannst du die Betriebskosten deines Spiels senken.

Auswahl der Architektur

GAMEPERF01: Wie wählst du die passende Hosting-Option für deine Spieleserver aus?

Die Auswahl der geeigneten Hosting-Option für Ihre Spieleserver ist für die Leistung Ihrer Spieleserver von grundlegender Bedeutung. Die Entscheidung, EC2 Instanzen, eine Container-

Lösung oder einen vollständig verwalteten Service zu verwenden, ist eine der ersten Entscheidungen, die bei der Architektur für die Produktion getroffen werden müssen. Jede Hosting-Option bietet unterschiedliche Funktionen und Überlegungen zur Leistungsoptimierung, Skalierung, zum Betrieb und zur Integration.

Best Practices

- [GAMEPERF01-BP01 Evaluieren Sie die Ressourcenanforderungen und Skalierbarkeitsanforderungen für Spieleserver](#)
- [GAMEPERF01-BP02 Berücksichtigen Sie den Betriebsaufwand für die Skalierung von Spieleservern](#)
- [GAMEPERF01-BP03 Evaluieren Sie die Integration mit anderen AWS Diensten, Entwicklungsumgebungen, Ziel-CPU-Architekturen und Funktionen](#)

GAMEPERF01-BP01 Evaluieren Sie die Ressourcenanforderungen und Skalierbarkeitsanforderungen für Spieleserver

Vergleichen Sie die Serveranforderungen mit Ihren Skalierbarkeitsanforderungen, um sicherzustellen, dass Sie eine Hosting-Option wählen, die sowohl Ihren Anforderungen entspricht als auch optimale Leistung bietet.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Berücksichtigen Sie bei der Auswahl der geeigneten Hosting-Option für Ihre Spieleserver die folgenden Faktoren:

Ressourcenanforderungen für Spieleserver

Beurteilen Sie die CPU-, Arbeitsspeicher-, Netzwerk- und Speicheranforderungen Ihrer Spieleserverprozesse, um festzustellen, wie viel Ihr Spiel verbraucht. Vergessen Sie nicht den Netzwerkbetrieb. Jeder Frame benötigt CPU-Zyklen, um Spieleraktionen zu empfangen, den Status des Spiels zu aktualisieren und die Daten an den Spieler zurückzuschicken. Durch das Auslagern der Paketverarbeitung kann die CPU für die Kernfunktionen des Spiels freigegeben werden. Netzwerke sind die Grundlage für einen reibungslosen und reaktionsschnellen Spielablauf. Wenn Sie also zu Beginn des Prozesses testen, wird ein grundlegendes Leistungsprofil für ein Spiel definiert.

Ein Ego-Shooter-Spiel kann hohe Aktionen pro Sekunde haben, die die CPU benötigt, um schnell ins Netzwerk zu wechseln, was rechenoptimierte Instanzen der C-Familie begünstigen kann, während ein rundenbasiertes Strategiespiel, bei dem mehr CPU-Zyklen pro Runde verarbeitet werden können, möglicherweise mehr Speicher von Instanzen der R-Familie benötigt, um den Status des Spiels lokal auf dem Server zu speichern und zu aktualisieren, bevor es an die Spieler zurückgesendet wird. Verwenden Sie einen datengesteuerten Ansatz wie [die Utilization Saturation and Errors \(USE\) - Methode](#), um fundierte Architekturentscheidungen zu treffen.

Skalierbarkeit und Elastizität

Prüfen Sie, wie schnell und reibungslos jede Hosting-Option skaliert werden kann, um den Anforderungen der Spieler gerecht zu werden, ohne die Leistung zu beeinträchtigen. Berücksichtigen Sie den Grad an Automatisierung und Flexibilität, der für die Arbeitslast Ihres Spiels erforderlich ist, um in Spitzenzeiten ein reibungsloses Spielerlebnis zu gewährleisten. Ein Spieleserver kann schnell skalieren, indem er die Auslastung erhöht, indem zusätzliche Spieleserverprozesse auf derselben Instanz hinzugefügt werden, wohingegen ein Spiele-Backend aufgrund der steigenden Anzahl aktiver Benutzer und der Anzahl der gespielten Spiele langsamer skaliert. Ihre Flotte sollte sich an die Nachfrage anpassen, um die Kosten zu minimieren und gleichzeitig die Wartezeit für die Spieler so gering wie möglich zu halten, bis sie mit dem Spiel beginnen können. Sehen Sie sich Amazon EC2 Spot Instance Advisor an, um einen Einblick in die kostengünstige verfügbare Kapazität für Spieleserverflotten zu erhalten.

Implementierungsschritte

- Evaluieren Sie die Anforderungen an die Spieleserverressourcen für CPU, Arbeitsspeicher, Netzwerk und Speicher, um geeignete Instance-Typen auszuwählen. Berücksichtigen Sie dabei spielspezifische Leistungsanforderungen wie einen hohen Netzwerkdurchsatz für FPS-Spiele oder Speicheroptimierung für rundenbasierte Strategiespiele.
- Vergleichen Sie verschiedene Hosting-Optionen wie Container, Instances, Bare-Metal und Managed Services, indem Sie Leistungsdaten mithilfe von Frameworks wie der USE-Methode analysieren. Nutzen Sie diese Erkenntnisse, um bessere Entscheidungen über Ihre Systemarchitektur zu treffen.
- Entwickeln Sie Flotten im Hinblick auf Skalierbarkeit und Elastizität und nutzen Sie Tools wie EC2 Spot Instance Advisor, um die Kosten zu optimieren und gleichzeitig eine schnelle Skalierung zu ermöglichen, um der Nachfrage der Spieler in Spitzenzeiten gerecht zu werden.

GAMEPERF01-BP02 Berücksichtigen Sie den Betriebsaufwand für die Skalierung von Spieleservern

Berücksichtigen Sie den Verwaltungs- und Betriebsaufwand, der mit jeder Hosting-Option verbunden ist.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Operativer Overhead

Selbst gehostete Lösungen auf EC2 oder in Containern können mehr Kontrolle bieten, erfordern aber auch mehr Management. Container-Orchestratoren wie ECS oder EKS können die Startzeiten für containerisierte Server reduzieren und gleichzeitig die Netzwerkkomplexität und den Aufwand für die Orchestrierung der Wartung erhöhen.

[EKS-verwaltete Knotengruppen](#) können beispielsweise die Bereitstellung und das Lebenszyklusmanagement Ihrer Spieleserver automatisieren, berücksichtigen aber nicht die Budgets für Pod-Unterbrechungen, wenn ein Knoten beendet wird. Wenn Ihr Spiel länger als die 15-minütige Kündigungsfrist benötigt, um Spiele sicher abzuschließen, müssen Sie möglicherweise Lifecycle-Hooks erstellen oder selbstverwaltete Knoten mit benutzerdefinierten Controllern in Betracht ziehen, um Spielunterbrechungen zu verhindern.

Managed Services wie Amazon Game Lift übernehmen zwar den größten Teil des Betriebsaufwands, reduzieren jedoch die Transparenz und Kontrolle über spezielle Anforderungen an Netzwerk- und Sicherheitskonfigurationen auf niedriger Ebene. Die Wahl einer Gameserverlösung ist ein Kompromiss zwischen dem Grad der Anpassung, der Kontrolle und der Verantwortung, die Sie für die Optimierung der Leistung des Spieleservers und das Skalierungsverhalten haben.

Implementierungsschritte

- Beurteilen Sie den betrieblichen Aufwand für Hosting-Optionen und wägen Sie den Kontroll- und Verwaltungsaufwand zwischen selbst gehosteten Lösungen wie EC2 ECS oder EKS und verwalteten Diensten wie Amazon Game Lift ab.
- Verwenden Sie verwaltete EKS-Knotengruppen für die Automatisierung, implementieren Sie jedoch Lifecycle-Hooks oder benutzerdefinierte Controller, wenn Ihre Spieleserver längere Kündigungsfristen als die Standardeinstellungen benötigen.

- Wägen Sie bei der Auswahl einer Spieleserverlösung die Kompromisse zwischen Anpassung, Transparenz und betrieblicher Verantwortung ab.

GAMEPERF01-BP03 Evaluieren Sie die Integration mit anderen AWS Diensten, Entwicklungsumgebungen, Ziel-CPU-Architekturen und Funktionen

Beurteilen Sie, wie gut sich jede Hosting-Option in andere AWS Dienste integrieren lässt, auf die Ihr Spiel angewiesen ist, z. B. Datenbanken, Analysen oder Dienste zur Bereitstellung von Inhalten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Integration in andere AWS -Services

Die nahtlose Integration zwischen den Diensten bietet betriebliche Vorteile wie eine verbesserte Leistungsüberwachung und eine effiziente, sichere Datenbereitstellung zwischen Spielekomponenten, Spiele-Backend-Diensten und Observability-Lösungen.

Beispielsweise kann die Koordination von Verkehrsverlagerungen bei Live-Spielen komplex sein. Amazon Route 53 hilft Ihnen dabei, Ihre DNS-Einträge auf dem neuesten Stand zu halten, was koordinierte Verkehrsverlagerungen vereinfacht. AWS Mit Global Accelerator-Traffic Dials können Sie einen Teil des Traffics in eine andere Region weiterleiten und Ihr Spiel während der Wartungsarbeiten am Laufen halten.

Entwicklungsumgebung und Tools

Berücksichtigen Sie die Entwicklungstools, Frameworks und Umgebungen, die von den einzelnen Architekturoptionen unterstützt werden. Stellen Sie sicher, dass die von Ihnen gewählte Option zu Ihrer Spieleentwicklungslösung und Ihren Programmiersprachen passt, da dies die Fähigkeit Ihres Teams beeinträchtigen kann, die Leistung des Spieleservers zu optimieren und aufrechtzuerhalten. Die Bereitstellung eines Spiels für Mobilgeräte, Konsolen und PCs erhöht die Komplexität der Tools und Tests. Systemübergreifender Support ist besonders wichtig für Studios mit mehreren Spielen, in denen zentralisierte Dienste bewährte Entwicklungsmethoden für alle Titel standardisieren können.

Architektur und Funktionen der Ziel-CPU

Berücksichtigen Sie das Leistungsprofil Ihrer Game-Engine- und Gameserver-Prozesse sowie den Grad der verfügbaren ARM-Unterstützung. Prüfen Sie, ob Sie von einem besseren Preis-Leistungs-

Verhältnis der ARM-basierten Graviton- oder x86-basierten AMD64 Prozessoren profitieren können. Müssen Sie Intel-Funktionen wie AES-NI-Verschlüsselung, AVX oder Turbo Boost verwenden? Sehen Sie [sich die Typen von Dedicated Hosts](#) an, um zwischen Instance-Familien mit einem oder mehreren Sockeln zu unterscheiden. Wenn Sie eine Instance-Familie mit mehreren Sockets verwenden, sollten Sie NUMA-Pinning und L3-Cache-Sharing bei Ihren Spielserversprozessen in Betracht ziehen. Verwenden Sie die [C-State- und P-State-Konfiguration](#), um die beste Leistung für Ihr Spiel zu erzielen, indem Sie die Taktfrequenz anpassen und den Ruhemodus reduzieren.

Implementierungsschritte

- Wählen Sie Hosting-Optionen mit nahtloser Integration mit AWS Diensten wie AWS Secrets Manager ACM und anderen, um die Leistungsüberwachung zu optimieren, die Datenbereitstellung zu sichern und manuelle Betriebsaufgaben zu reduzieren.
- Überprüfen Sie die Kompatibilität zwischen Ihrer Hosting-Option und Ihrer Entwicklungsumgebung, Frameworks und Programmiersprachen, um die Serverleistung effektiv zu optimieren und aufrechtzuerhalten.
- Evaluieren Sie die Anforderungen an die CPU-Architektur, nutzen Sie Graviton für ein gutes Preis-Leistungs-Verhältnis oder x86 für spezifische Funktionen wie AES-NI, AVX und Turbo Boost und optimieren Sie die Serverleistung mit NUMA-Pinning und C-State/P-State-Tuning.

Auswahl der Region

GAMEPERF02: Wie bestimmen Sie, in welchen geografischen Regionen Ihre Spieleinfrastruktur gehostet werden soll?

Die Wahl des idealen Standorts für Ihre Spieleinfrastruktur kann die Netzwerkleistung für Spieler und Backend verbessern. Es ist wichtig für langfristiges Wachstum und Nachhaltigkeit in einer geografischen Region zu berücksichtigen, von wo aus sich Ihre Spielerbasis verbindet und wie Ihre Communitys oder Server aufgebaut sind. Der Einsatz entkoppelter Spielservers-Infrastruktur und Backend-Services kann deine betriebliche Gesamteffizienz verbessern und die Ausfallsicherheit verbessern, indem du mehrere Regionen, lokale Zonen und Außenposten für das Hosten deines Spiels nutzt.

Best Practices

- [GAMEPERF02-BP01 Wähle eine Heimatregion in der Nähe deiner Spieler](#)

- [GAMEPERF02-BP02 Entwickeln Sie einen Ansatz, der die Platzierung latenzempfindlicher Spielinfrastrukturen in der Nähe der Spieler unterstützt, um die Leistung zu verbessern](#)

GAMEPERF02-BP01 Wähle eine Heimatregion in der Nähe deiner Spieler

Bei der ersten Veröffentlichung eines Spiels sollten Sie anhand von Gesprächen mit Ihren Geschäftspartnern festlegen, wie z. B. den Veröffentlichungsteams, die festlegen, wo das Spiel den Spielern voraussichtlich zur Verfügung gestellt wird und worauf sie ihre Marketing- und Werbemaßnahmen vor der Veröffentlichung konzentrieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Ihre Unternehmensvertreter sollten auch über Mechanismen verfügen, um die Nachfrage anzukurbeln und so ein besseres Verständnis der Akzeptanz und Rentabilität der Spieler zu gewinnen. Diese Teams werden zum Beispiel über Mechanismen wie Vorbestellungen von Spielen, Marketingveranstaltungen und Kampagnen, öffentliche E-Mail-Listen, über die Spieler vor dem Start ihr Interesse anmelden können, und andere Methoden einsetzen, um relevante Signale zu setzen, anhand derer bestimmt werden kann, wo das Spiel bei der Veröffentlichung wahrscheinlich die meisten Spieler haben wird. Das Spiel kann auch eine regionale Einführungsstrategie verfolgen, die Testphasen und Soft-Launch-Phasen umfasst, um die regionale Nachfrage der Spieler zu ermitteln.

[Wählen Sie eine Heimatregion](#) aus, die sich in der Nähe Ihrer Spielerbasis und Ihrer Entwickler befindet und die AWS Dienste und Funktionen bietet, die Sie für das Hosten Ihres Spiels benötigen. Das Zuhause RRegion wird der Ort sein, an dem die Backend-Dienste des Spiels ausgeführt werden, und dort können auch Spieleserver laufen. Beurteilen Sie eine Heimatregion anhand der unterstützten Dienste, der Konnektivität zu Edge-Standorten, der Nähe zu Failover-Regionen und der Anzahl der Availability Zones. Wenn Sie eine lokale Zone verwenden, sollten Sie berücksichtigen, dass sich die übergeordnete Region manchmal in einem anderen geografischen Gebiet befindet. Ein Beispiel: Die lokale Zone us-east-1-scl-1a in Santiago, Chile, hat Nord-Virginia us-east-1 als übergeordnete Region, obwohl sie geografisch näher an Sao Paulo sa-east-1 liegt.

Implementierungsschritte

- Identifizieren Sie die Einsatzregionen anhand von Spielernachfragesignalen aus Aktivitäten vor der Markteinführung wie Vorbestellungen, Marketingkampagnen und Anmeldungen von Interessenten.

- Wählen Sie eine Heimatregion in der Nähe der primären Spielerbasis und der Entwickler und stellen Sie sicher, dass sie die erforderlichen AWS Dienste, Edge-Standorte und Failover-Regionen unterstützt.
- Prüfen Sie die Local Zones sorgfältig, da sich die übergeordnete Region geografisch vom Standort der lokalen Zone unterscheiden kann.

GAMEPERF02-BP02 Entwickeln Sie einen Ansatz, der die Platzierung latenzempfindlicher Spielinfrastrukturen in der Nähe der Spieler unterstützt, um die Leistung zu verbessern

Durch die separate Platzierung latenzempfindlicher Infrastrukturen wie Spieleserver werden die Auswirkungen langer Netzwerkrouuten minimiert. Wiederholbare Bereitstellungen können es einfach machen, mehrere Standorte zu verwalten, die für Ihre Spieler leistungsfähiger sind. Ping ist eine gängige Metrik, die in der Benutzeroberfläche des Spiels auftaucht, und ein niedriger Ping kann ein Unterscheidungsmerkmal sein.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Wenn Sie ein Spiel zum ersten Mal starten, verfügen Sie möglicherweise noch nicht über genügend Informationen über Ihre Spielerbasis, um ausreichend zu wissen, wo Sie die Infrastruktur am besten einsetzen können, die den Spielern am nächsten ist, die Ihr Spiel am meisten spielen möchten. Dies ist eine häufige Herausforderung, und Sie sollten sich auf dieses Szenario vorbereiten, indem Sie eine Architektur entwerfen, die es Ihnen ermöglicht, Ihre Hosting-Platzierungsstrategie schnell anzupassen, um Server dort zu platzieren, wo sie in der Nähe der Spieler benötigt werden. Es ist typisch für Spieleentwickler, die Bereitstellung ihrer Spieleinfrastruktur regelmäßig als wiederkehrende Analyse nach der Veröffentlichung zu bewerten, um im Laufe der Zeit schrittweise in Verbesserungen mit einem iterativen Ansatz zu investieren.

Eine bewährte Methode besteht darin, infrastructure-as-code Vorlagen wie AWS CloudFormation oder Terraform von Hashicorp für die Konfiguration Ihrer Infrastruktur zu verwenden, z. B. VPCs für Subnetzkonfigurationen und Abhängigkeiten, die für den Start kritischer Spieledienste erforderlich sind, sodass Sie auf diese Vorlagen zurückgreifen, sie bei Bedarf schnell anpassen und sie an Orten einsetzen können, an denen zusätzliche Infrastruktur zur Unterstützung Ihrer Spieler benötigt wird.

Sie sollten auch sicherstellen, dass Sie verstehen, wie Ihre aktuelle Bereitstellungsstrategie weiterentwickelt werden könnte, um future Erweiterungen zu ermöglichen. IaC-Vorlagen sind

wiederholbar, stellen jedoch keinen Ersatz für die Netzwerkplanung dar. [IPAM verwaltet Ihre](#). VPCs Subnetzdimensionierung, Auswahl der Availability Zone und IP-Inventar sowie kontoübergreifende Ausrichtung der Availability Zone. Es ist wichtig, das Netzwerk zu berücksichtigen, und es kann für Spieler störend sein, wenn es geändert wird. Spieleserver, die an mehreren geografischen Standorten installiert sind, stellen eine Verbindung zu deinem Spiel-Backend her. Üblicherweise werden sie in einer oder mehreren Heimatregionen gehostet, was eine zusätzliche Konfiguration erfordern kann, um private Konnektivität zu unterstützen. Diese Überlegungen sollten im Laufe der Zeit kontinuierlich überprüft werden, sodass Sie Änderungen an Ihrer Strategie für das Hosten von Spielen vornehmen können, wenn sich die Anforderungen Ihres Spiels oder Ihre Spielieranforderungen ändern.

Wenn du festlegst, wie viele Spielhosting-Standorte du für dein Spiel nutzen möchtest, solltest du die folgenden Faktoren berücksichtigen:

- Verbesserung der Qualität des Spielererlebnisses: Wie stark können Sie das Spielerlebnis verbessern, indem Sie zusätzliche Standorte hinzufügen, an denen Spiele gehostet werden? Was ist der inkrementelle Leistungszuwachs, den Sie auf diese Weise erzielen können? Wie werden Sie diese Leistungsverbesserung messen?
- Welche Spielerpopulationen sollten priorisiert werden: Für wie viele Spieler kannst du das Spielerlebnis verbessern, wenn du zusätzliche Standorte für Spielveranstalter hinzufügst? Welchen Spielerpopulationen oder geografischen Standorten werdet ihr Priorität einräumen?
- Auswirkungen des Wandels im Nachhinein: Wenn du deine Strategie für das Hosten von Spielen änderst, wie wird sich das auf deine Wartezeiten beim Matchmaking für Spieler auswirken? Können die Größe der Spiele, die Spielstärke oder die Anzahl der Spieler im Spielerpool einer Änderung der Strategie für den Austragungsort eines Spiels Rechnung tragen? Die Unterstützung weiterer Standorte kann den Spielerpool potenziell fragmentieren und die Kosten und Komplexität erhöhen.

Jede dieser Überlegungen sollte geprüft werden, wenn Sie entscheiden, wo Sie Standorte für Spiele-Hostings hinzufügen oder entfernen möchten. Sie können sich beispielsweise dafür entscheiden, die Verbesserung des Spielerlebnisses für Spieler an geografischen Standorten mit der schlechtesten Spielerfahrung oder für Spieler, die das lautstärkste öffentliche Feedback äußern, zu priorisieren. Sie können sich auch dafür entscheiden, die Monetarisierung von Spielern in Ihre Prioritäten einzubeziehen, indem Sie sich beispielsweise darauf konzentrieren, das Spielerlebnis für Spieler an geografischen Standorten zu verbessern, die eine bedeutende Einnahmequelle für Ihr Spiel darstellen oder die das Potenzial haben, zusätzliche Einnahmen zu erzielen, wenn Sie Leistungsverbesserungen einführen.

Zusätzlich zum Hosten der Infrastruktur in AWS-Regionen können Sie [Local Zones](#) verwenden, die eine Erweiterung von sind AWS-Region, um Ihre Spieleserver und andere latenzempfindliche Anwendungen wie Voice-Chat-Server näher an Ihren Spielern zu hosten. Sie können sich auch dafür entscheiden, die Infrastruktur für die Spieleentwicklung in Local Zones zu betreiben, um das Spielerlebnis für Ihre Spieleentwicklungsteams zu verbessern. Sie können Local Zones beispielsweise verwenden, um Anwendungsfälle wie das Hosten von Replikaten Ihrer selbst verwalteten Quellcodeverwaltungsserver näher an Ihren Spieleentwicklern zu adressieren und Benutzern, die EC2 Amazon-Instances, EBS-Volumes und FSx Amazon-Dateisysteme verwenden, die in einer oder mehreren Local Zones in der Nähe Ihrer Entwicklungsstudios installiert sind, virtuelle Workstations und Inhaltsspeicher für die Spieleentwicklung anzubieten, ohne dass Sie die Infrastruktur vor Ort hosten müssen.

[Outposts](#) sind eine gute Wahl, wenn Regionen oder Local Zones nicht in demselben geografischen Gebiet verfügbar sind. Die Konnektivität zwischen deinem Rechenzentrum und dem Backend-System AWS sollte in Betracht gezogen werden, um die Zuverlässigkeit des Spieleservers zum Backend-System zu gewährleisten. AWS Outposts und Outpost Server wurden speziell für den Betrieb AWS in Ihrem Rechenzentrum mit denselben Diensten entwickelt und helfen Ihnen, ein einheitliches Bereitstellungsmodell APIs zu erstellen, wo auch immer Sie Ihr Spiel ausführen. Mehrere Racks können zu einem logischen Outpost kombiniert werden, und die Infrastruktur kann gemeinsam genutzt werden. AWS-Konten Der Hardware-Lebenszyklus wird von verwaltet AWS und die Vorlaufzeit kann bis zu 3 Monate betragen.

Wenn Sie Spiele mit Containern entwickeln und die Flexibilität haben möchten, eine hybride Bereitstellungsarchitektur mit Open-Source-Software einzuführen, die auf Ihrer eigenen lokalen Infrastruktur bereitgestellt werden kann, können Sie [ECS Anywhere oder EKS Anywhere](#) als Alternative zu AWS Outposts oder Local Zones verwenden. Wenn Sie bei Amazon hosten GameLift, [kann Amazon GameLift Anywhere verwendet werden, um Ihren Server auf lokaler Hardware auszuführen](#), was Ihren Entwicklungsprozess beschleunigen kann, sodass Sie Local Zones verwenden oder Ihr eigenes Metall als Teil Ihrer Flotte registrieren können.

Implementierungsschritte

- Verwenden Sie infrastructure-as-code Tools wie AWS CloudFormation oder Terraform für wiederholbare Bereitstellungen, die eine schnelle Anpassung und Skalierung der Hosting-Standorte für Spiele an die Bedürfnisse der Spieler ermöglichen.
- Beurteilen Sie die Verbesserung der Spielererfahrung, die Prioritäten der Spielerpopulation und nachgelagerte Auswirkungen wie die Matchmaking-Zeiten beim Hinzufügen oder Entfernen von Spielveranstaltungsorten.

- Verwenden Sie AWS Local Zones, Outposts oder Hybridoptionen wie ECS Anywhere, EKS Anywhere oder GameLift Anywhere, um latenzempfindliche Infrastrukturen zu optimieren und unterschiedliche Bereitstellungsanforderungen zu unterstützen.

Iterative Entwicklung

GAMEPERF03: Wie können Sie Amazon GameLift für die Leistungseffizienz iterativer Entwicklungen nutzen?

Amazon GameLift bietet einen end-to-end Workflow für die Entwicklung und Leistungstests Ihres Spiels in einer lokalen Testumgebung.

Best Practices

- [GAMEPERF03-BP01 Verwenden Sie Amazon GameLift Anywhere und ein Test-Toolkit GameLift](#)
- [GAMEPERF03-BP02 Testen Sie die Leistung und Skalierbarkeit von Spieleservern](#)
- [GAMEPERF03-BP03 Optimieren Sie die Ressourcennutzung von Containern GameLift](#)

GAMEPERF03-BP01 Verwenden Sie Amazon GameLift Anywhere und ein Test-Toolkit GameLift

Um die Leistungseffizienz durch einen iterativen Entwicklungsprozess zu verbessern, nutzen Sie Amazon GameLift Anywhere zusammen mit dem Amazon GameLift Testing Toolkit, um eine umfassende Testumgebung einzurichten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Dieser Ansatz ermöglicht eine schnelle Iteration, eine effiziente Datenerfassung und eine detaillierte Leistungsanalyse. Zu den wichtigsten Schritten gehören:

Erstellen Sie eine Testumgebung

Verwenden Sie Amazon GameLift Anywhere, um eine lokale oder cloudbasierte Testumgebung einzurichten. Durch dieses Setup entfällt die Notwendigkeit, jede Spielservers-Build-Iteration auf eine verwaltete Flotte hochzuladen, wodurch die Aktivierungszeit reduziert wird.

Integrieren Sie das Amazon GameLift Testing Toolkit

Integrieren Sie das Amazon GameLift Testing Toolkit in Ihren Entwicklungsablauf. Das Toolkit bietet Skripte, Tools und Bibliotheken zur Visualisierung der GameLift Amazon-Infrastruktur, zum Starten virtueller Player und zum Iterieren von FlexMatch Regelsätzen mit dem FlexMatch Simulator. Es vereinfacht die Integration und Verwaltung von GameLift Amazon-Ressourcen und ermöglicht es Ihnen, allgemeine Aufgaben zu automatisieren und die für die Leistungsanalyse erforderlichen Daten zu sammeln.

Schnelle Bau- und Testzyklen

Aktualisieren Sie die Testflotte schnell mit neuen Builds, starten Sie sie und beginnen Sie mit den Tests. Dies ermöglicht einen schnellen build-test-repeat Zyklus und ermöglicht es Entwicklern, verschiedene Aspekte des Spielerlebnisses, einschließlich Multiplayer-Interaktionen, zu überprüfen.

Umfassende Tests

Testen Sie Ihre Gameserver-Integration mit dem Amazon GameLift Server SDK, Backend-Serviceinteraktionen, Matchmaking-Konfigurationen und anderen GameLift Hosting-Funktionen. Verwenden Sie das GameLift Testing-Toolkit, um Tests zu automatisieren und detaillierte Leistungskennzahlen zu sammeln, um sicherzustellen, dass die Spielkomponenten reibungslos zusammenarbeiten.

Analysieren Sie Leistungsdaten

Verwenden Sie die vom GameLift Testing Toolkit gesammelten Daten, um Leistungsengpässe zu analysieren und Ihren Spielserverserver zu optimieren. Das Toolkit hilft dabei, wichtige Kennzahlen zu verfolgen, Probleme zu identifizieren und datengestützte Entscheidungen zur Verbesserung der Leistungseffizienz zu treffen.

Durch die Integration von Amazon GameLift Anywhere und dem GameLift Testing Toolkit in Ihren iterativen Entwicklungsprozess können Sie die Leistungseffizienz durch schnelle Tests, umfassende Integrationsprüfungen und detaillierte Leistungsanalysen erheblich verbessern.

Implementierungsschritte

- Verwenden Sie Amazon GameLift Anywhere, um eine Testumgebung zu erstellen, die die Aktivierungszeit für Spieleserver-Builds reduziert und eine schnelle Iteration ermöglicht.
- Integrieren Sie das Amazon GameLift Testing Toolkit, um Testaufgaben zu automatisieren, Spieler zu simulieren und FlexMatch Konfigurationen während der Entwicklung zu validieren.
- Sammeln und analysieren Sie Leistungsdaten mit dem GameLift Testing Toolkit, um Engpässe zu identifizieren, Spieleserver zu optimieren und die Leistungseffizienz zu verbessern.

GAMEPERF03-BP02 Testen Sie die Leistung und Skalierbarkeit von Spieleservern

Um die Leistung und Skalierbarkeit Ihrer Spieleserver zu testen, implementieren Sie ein robustes Test-Framework mit den GameLift Amazon-Funktionen und dem GameLift Testing Toolkit.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Zu den wichtigsten Praktiken gehören:

Iteratives Testen

Verwenden Sie eine Amazon GameLift Anywhere-Flotte, um eine cloudbasierte, gehostete Umgebung zu erstellen, in der Sie Spielkomponenten iterativ erstellen und testen können. Diese Umgebung sollte die realen Hosting-Bedingungen widerspiegeln und realistische Leistungs- und Skalierbarkeitstests ermöglichen.

Testen der Integration von Spieleservern

Testen Sie die Integration Ihres Spieleservers mit dem Amazon GameLift Server SDK, einschließlich des Starts neuer Spielsitzungen und der Nachverfolgung von Spielsitzungen mithilfe unseres AWS CLI GameLift Testing-Toolkits. Dadurch wird überprüft, ob der Spieleserver in der GameLift Umgebung ordnungsgemäß funktioniert.

Verwende das GameLift Testing-Toolkit, um Tests zu automatisieren und detaillierte Leistungskennzahlen zu sammeln. Mit dem Toolkit können Sie die GameLift Infrastruktur visualisieren, virtuelle Player für Auslastungstests starten und FlexMatch Regelsätze mit dem Simulator iterieren. FlexMatch Es ist besonders nützlich für die Skalierung von ECS

Fargate-Aufgaben, bei denen Spilersitzungen simuliert werden, indem zahlreiche gleichzeitige Spielsitzungen erstellt werden, um die Serverinfrastruktur einem Stresstest zu unterziehen.

Skalierbarkeitstests

Experimentieren Sie mit Warteschlangendesigns für Spielsitzungen, Flotten mit mehreren Standorten, Spot- und On-Demand-Flotten und mehreren Instance-Typen. Testen Sie die Platzierungsoptionen für Spielsitzungen, Latenzrichtlinien und Einstellungen für die Flottenpriorisierung. Konfigurieren Sie die Kapazitätsskalierung, um der Nachfrage der Spieler gerecht zu werden, und überprüfen Sie, ob das System die erwartete Last unter verschiedenen Bedingungen bewältigen kann.

Implementierungsschritte

- Verwenden Sie Amazon GameLift Anywhere, um eine realistische Testumgebung für iterative Leistungs- und Skalierbarkeitstests einzurichten.
- Testen Sie die Gameserver-Integration mit dem GameLift Server-SDK, um die korrekte Sitzungsverwaltung und die Ereignisverfolgung innerhalb der GameLift Umgebung zu erleichtern.
- Führen Sie Skalierbarkeitstests mit dem GameLift Testing Toolkit durch, simulieren Sie die Auslastung der Spieler, testen Sie Sitzungswarteschlangen und validieren Sie die Skalierung der Flotte, die Latenzrichtlinien und die Priorisierungseinstellungen.

GAMEPERF03-BP03 Optimieren Sie die Ressourcennutzung von Containern GameLift

Um die Ressourcennutzung von GameLift Containern zu optimieren, sollten Sie Ihre Containerflotte effektiv gestalten und präzise Ressourcenlimits festlegen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Zu den wichtigsten Richtlinien gehören:

- Design von Containergruppen: Organisieren Sie Ihre Software in Containergruppen. Der primäre Container sollte deine Gameserver-Anwendung und den Amazon GameLift Agent bündeln. Verwenden Sie Sidecar-Container für zusätzliche Software, um Abhängigkeiten zu verwalten und containerspezifische Grenzwerte für die Speicher- und CPU-Auslastung festzulegen.
- Legen Sie Ressourcenlimits fest: Ermitteln Sie für jede Containergruppe die erforderlichen Speicher- und CPU-Ressourcen. Legen Sie optionale Grenzwerte für einzelne Container fest, um

sicherzustellen, dass sie über reservierte Ressourcen verfügen. Sie können diese Grenzwerte jedoch auch überschreiten, wenn zusätzliche Ressourcen verfügbar sind. Dies trägt dazu bei, Ressourcenkonflikte und mögliche Containerausfälle zu verhindern.

- **Daemon-Containergruppe:** Erwägen Sie die Verwendung einer Daemon-Containergruppe für Hintergrund- oder Überwachungsprozesse, die nicht mit der primären Containergruppe skaliert werden müssen. Dadurch wird sichergestellt, dass wichtige Hintergrundaufgaben effizient erledigt werden, ohne die Prozesse des primären Spieleservers zu beeinträchtigen.

Implementierungsschritte

- Entwerfen Sie Containergruppen mit einem primären Container für den Spieleserver und den GameLift Agenten und Sidecars für die Verwaltung von Abhängigkeiten mit spezifischen Speicher- und CPU-Limits.
- Lege Ressourcenlimits für jede Containergruppe fest, um benötigte Ressourcen zu reservieren und gleichzeitig eine kontrollierte Ressourcennutzung zu ermöglichen, um Konflikte zu vermeiden.
- Verwenden Sie eine Daemon-Containergruppe für Hintergrund- oder Überwachungsaufgaben und stellen Sie sicher, dass sie effizient funktionieren, ohne die primären Spieleserverprozesse zu beeinträchtigen.

Computer und Hardware

GAMEPERF04: Wie verhindert man, dass Spielsitzungen Auswirkungen auf Spieler haben, die auf derselben Spieleserver-Instance laufen?

Nachdem Ihr Spieleserver in Betrieb genommen wurde AWS, müssen Sie dessen Leistung überwachen, um unabhängig von der Ressourcenauslastung, der zugrunde liegenden Rechenleistung oder der Auslastung ein qualitativ hochwertiges Spielerlebnis zu bieten.

Best Practices

- [GAMEPERF04-BP01 Überwachen Sie die Spieleserverprozesse, um Probleme zu erkennen](#)
- [GAMEPERF04-BP02 Leistungstest deinen Gameserver mit simulierten und realen Gameplay-Szenarien](#)

GAMEPERF04-BP01 Überwachen Sie die Spielseverprozesse, um Probleme zu erkennen

Sie können mehrere Spielseverprozesse pro Instanz ausführen, um die Ressourcen auf Ihren Spielsever-Instanzen effizient zu nutzen. Wenn ja, entwerfen Sie Ihre Architektur so, dass ein einzelner Spielseverprozess, der eine Spielsitzung hostet, keine negativen Auswirkungen auf andere Spielsitzungen haben kann, die auf derselben Instanz gehostet werden. Ermitteln Sie anhand von Kennzahlen, wie sich Spielplatzierung und Spielmodusart auf die Leistung von Spielsever-Instanzen auswirken können. Integrieren Sie eine Mischung aus Prozessen mit geringer Auslastung (Lobby, Shop oder Einzelspieler-Tutorial) und Prozessen mit hoher Auslastung (Ranglisten-, Mehrspieler- oder High-Skill-Gameplay), um zu verhindern, dass die Spielseverinstanz Hotspotting bekommt.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Überwachen Sie das Spielerlebnis anhand von client- und serverseitigen Metriken, indem Sie Telemetriedaten zu Ping-Zeit und Jitter, Framedrops, API-Reaktionszeit, Fehlern und erfolgreichem Abschluss der Spielschleife sammeln. Korrelieren Sie die Zeitstempel für diese Ereignisse mit Problemen beim Spielersupport und Serverprotokollen, um Leistungsengpässe zu identifizieren. Tools wie [Dtrace](#), [ftrace](#), [uperf](#) und [eBPF](#) können für eine gründliche Untersuchung und Analyse der Systemleistung verwendet werden.

Implementieren Sie die Überwachung der begrenzten Ressourcen, die Ihren Spielsever-Instanzen zur Verfügung stehen, sodass Sie Warnmeldungen ausgeben können, wenn einzelne Spielseverprozesse vordefinierte Schwellenwerte für das Ressourcenbudget überschreiten. Wenn Schwellenwerte überschritten werden, sollten Sie Ihre Gameserver-Software so konfigurieren, dass relevante System- und Spielsever-Abmeldungen in einem dauerhaften Speicher abgelegt werden, z. B. in einer zentralen Protokollierungslösung, sodass Ihre Gameserver-Techniker dieses Verhalten untersuchen können. Darüber hinaus sollte Ihre Spielsever-Instanz so konfiguriert sein, dass Messwerte von jedem der auf der Instance ausgeführten Spielseverprozesse gemeldet werden, sodass Sie diese einzelnen Spielseverprozesse zusätzlich zu den Gesamtmetriken für die Spielseverinstanz überwachen können.

GameLift Bietet beispielsweise Metriken für die [Überwachung von Spielsitzungen](#), die durch benutzerdefinierte spielspezifische Metriken und Protokolle erweitert werden können, die mit dem [Amazon CloudWatch Agent](#) erfasst wurden und den Sie auf Ihrer Spielsever-Instance

konfigurieren können. Ihre Metriken können in anderen Tools wie [Amazon Managed Grafana](#) angezeigt CloudWatch oder in diese exportiert werden, das in Single Sign-On integriert ist, um Benutzern, die möglicherweise keinen Zugriff auf die Management Console haben, den Zugriff auf Metriken zu erleichtern. Beachten Sie die folgenden bewährten Methoden für die [Verwaltung von Protokollen und Metriken mit Amazon GameLift](#), das auch Unterstützung für die Anzeige einzelner [Spielsitzungsprotokolle](#) bietet.

Implementierungsschritte

- Führen Sie mehrere Spielserverprozesse pro Instanz aus und kombinieren Sie Spielmodi mit niedriger und hoher Auslastung, um Hotspotting zu vermeiden und eine ausgewogene Ressourcennutzung sicherzustellen.
- Überwachen Sie client- und serverseitige Messwerte wie Ping, Jitter, Framedrops und API-Antwortzeiten und korrelieren Sie diese mit Serverprotokollen und Problemen, die von Spielern gemeldet wurden, um Engpässe zu identifizieren.
- Konfigurieren Sie die Ressourcenüberwachung für jeden Spielserverprozess, generieren Sie Benachrichtigungen bei Schwellenwertverletzungen und speichern Sie Protokolle zur Analyse in einem dauerhaften Speicher mit Tools wie CloudWatch Amazon Managed Grafana.

GAMEPERF04-BP02 Leistungstest deinen Gameserver mit simulierten und realen Gameplay-Szenarien

Führen Sie Leistungstests durch und bewerten Sie verschiedene Spielszenarien, um festzustellen, ob der Spielserverprozess die Nutzung fester Ressourcen wie EC2 Instanzspeicher, CPU und Netzwerkbandbreite angemessen handhabt.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Durch die Erstellung simulierter Gameplay-Tests mit Bots, die gängige Gameplay-Pfade und Verhaltensweisen deiner Spieler widerspiegeln können, lässt sich ermitteln, wie deine Gameserver-Prozesse dies in verschiedenen Nutzungsszenarien handhaben. Sie können beispielsweise eine Lösung wie [Distributed Load Testing](#) implementieren, die Sie so anpassen können AWS, dass Game-Client-Simulationen oder Game-Client-Builds zur Generierung von Spielszenarien ausgeführt werden. Führen Sie interne Spieltests durch und setzen Sie QA-Teams ein, um die verschiedenen Funktionen Ihres Spiels einem Stresstest zu unterziehen, sodass Sie sich darauf verlassen können, dass Ihr

Spiel optimal funktioniert. [AWS Device Farm](#) kann verwendet werden, um Mobil- und Webtests für Ihre iOS-, Android- und Browsergames auf mehreren Gerätetypen durchzuführen.

Implementierungsschritte

- Führen Sie Leistungstests mit Bots durch, die das allgemeine Spielerverhalten simulieren, um die Auslastung der Spieleserverressourcen in verschiedenen Szenarien zu bewerten.
- Verwende Lösungen wie Distributed Load Testing AWS , um Gameplay-Szenarien für Stresstests anzupassen und zu simulieren.
- Führe interne Spieltests durch und verwende Tools AWS Device Farm zum Testen von Mobil- und Browserspielen auf verschiedenen Geräten.

Auswahl berechnen

GAMEPERF05: Wie wählst du die passende Rechenlösung für dein Spiel aus?

Die Rechenleistung variiert je nach Instanzgröße und Instance-Familie. Es ist vorteilhaft, mehrere Rechenoptionen zu verwenden, die aus separaten Kapazitätspools stammen. Entwickeln Sie eine Strategie zur Flottenzusammensetzung, die der Leistung den Vorzug gibt, aber auch genügend Vielfalt beinhaltet, um Fehler bei unzureichender Kapazität zu vermeiden.

Best Practices

- [GAMEPERF05-BP01 Benchmarkiere deine Spieleleistung auf mehreren Computertypen](#)
- [GAMEPERF05-BP02 Verschieben Sie non-latency-sensitive Rechenaufgaben in asynchrone Workflows](#)

GAMEPERF05-BP01 Benchmarkiere deine Spieleleistung auf mehreren Computertypen

Für Gameserver-Workloads gibt es keinen einzigen Ansatz, um die optimale Rechenlösung für das Hosten Ihres Spieleservers zu finden. Eine gängige Strategie für das Benchmarking von Spieleservern besteht darin, mit rechenoptimierten EC2 C-Instances zu beginnen, da diese Instance-Familie eine hohe Leistung für rechenintensive Workloads bietet. Wenn Ihr Spiel jedoch viel Speicher

benötigt, um bestimmte Funktionen zu implementieren, sind die speicheroptimierten Instances möglicherweise am besten geeignet.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Wenn Ihre Arbeitslast erhebliche Netzwerkressourcen beansprucht, sollten Sie in Erwägung ziehen, netzwerkoptimierte Instances zu implementieren, was normalerweise mit einem „n“ im Instanznamen gekennzeichnet ist. Vermeiden Sie instabile Instance-Typen „t“, da die Leistung nach Erschöpfung der Credits abnimmt. Spiele reagieren empfindlich auf Latenz und verworfene Pakete. Es wird daher empfohlen, EC2 Enhanced Networking zu verwenden, um die Netzwerkleistung Ihrer Spieleserver zu verbessern. [Enhanced Networking verwendet Single Root I/O Virtualization \(SR-IOV\), um leistungsstarke Netzwerkfunktionen für unterstützte Instance-Typen bereitzustellen](#). SR-IOV ist eine Methode der Gerätevirtualisierung, die im Vergleich zu herkömmlichen virtualisierten Netzwerkschnittstellen eine höhere I/O Leistung und eine geringere CPU-Auslastung bietet. Die optimierte Netzwerkleistung ermöglicht eine größere Bandbreite, mehr Pakete pro Sekunde (PPS) und konstant niedrigere Latenzzeiten zwischen Instances. [Enhanced Networking mit Elastic Network Adapter ist für die neuesten EC2 Instance-Typen verfügbar. Es ist wichtig, dass regelmäßig Updates durchgeführt werden, um von den Leistungsverbesserungen neuerer Instances und den Verbesserungen des Nitro-Hypervisors zu profitieren.](#)[AWS](#)

Wenn Ihr Spiel über mehrere EC2 Instanztypen hinweg eine ähnliche Leistung erbringt, sollten Sie erwägen, mehrere Instanztypen zum Hosten Ihrer Spieleserver zu verwenden. Überwachen Sie die Leistung im Laufe der Zeit und führen Sie weitere Optimierungen durch, nachdem Sie genügend Spielesitzungen in der Produktionsumgebung veranstaltet haben, um Leistungstrends erkennen zu können. Denken Sie daran, dass sich Ihre Rechenanforderungen ändern können, wenn Sie Ihrem Spiel neue Funktionen hinzufügen, die eine andere Ressourcenzuweisung erfordern. Sie können [EC2 Auto Scaling Scaling-Gruppen so konfigurieren](#), dass sie mehrere Instanztypen verwenden, oder Sie können separate Auto Scaling Scaling-Gruppen verwenden, um Spieleserver-Instanzen zu hosten, die separate Instanztypen ausführen, was die Verwaltung der Korrelation und Aggregation von Metriken möglicherweise vereinfacht.

Beurteilen Sie, wie Ihr Spiel auf verschiedenen Prozessortypen wie Intel-basierten Instances, AMD-basierten Instances und ARM-basierten Graviton-Instances abschneidet. Unreal Engine 5.1.1 oder [neuer kann Spieleserver für Graviton kompilieren und das Preis-Leistungs-Verhältnis Ihres Spiels](#) verbessern. Führen Sie Sweep- und Sättigungstests bei verschiedenen Größen innerhalb jeder Produktreihe durch, um den optimalen Punkt zu ermitteln, an dem Auslastung und Leistung konsistent sind.

Sie sollten auch überprüfen, wie sich die Leistung Ihres Spiels auswirkt, wenn es mithilfe von Containern und Lambda-Funktionen gehostet wird. Für Anwendungsfälle, in denen langlebige Spieleserverprozesse nicht erforderlich sind, wie z. B. asynchrone Spiele und Backend-Dienste für Spiele, sollten Sie die Verwendung einer serverlosen Architektur mit Lambda in Betracht ziehen, die die Verwaltung und den Betrieb für Spielbetriebsteams vereinfachen und es Ihnen ermöglicht, Ihr Spiel schneller global für viele bereitzustellen. AWS-Regionen Bewährte Methoden für serverlose [Anwendungen finden Sie im Serverless Applications Lens — Well-Architected](#) Framework.

Implementierungsschritte

- Vergleichen Sie Spieleserver anhand von rechenoptimierten C-Instances für CPU-intensive Workloads, speicheroptimierten Instances für speicherintensive Aufgaben und netzwerkoptimierten N-Instances für hohen Netzwerkdurchsatz.
- Verwenden Sie Enhanced Networking mit Elastic Network Adapter (ENA) auf unterstützten Instances, um die Netzwerkleistung zu verbessern, die Latenz zu reduzieren und die Paketverarbeitungsrate zu erhöhen.
- Evaluieren und testen Sie mehrere Instance-Typen, Prozessoren (Intel, AMD, Graviton) und Container- oder Lambda-Hosting-Optionen und passen Sie die Rechenlösungen an die Weiterentwicklung der Spielfunktionen an.

Weitere Informationen finden [Sie unter Wählen Sie die richtige Rechenstrategie für Ihre globalen Spieleserver](#).

GAMEPERF05-BP02 Verschieben Sie non-latency-sensitive Rechenaufgaben in asynchrone Workflows

Wenn Sie die Leistung Ihres Spiels optimieren, sollten Sie bedenken, dass nur einige der Interaktionen zwischen dem Client und dem Spiel-Backend synchron ausgeführt werden müssen. Sie sollten jedes Feature aus der Perspektive der Spielererfahrung betrachten und entscheiden, ob bestimmte Interaktionen synchrone Kommunikation erfordern, die blockierend und ressourcenintensiv ist, oder ob diese Funktionen asynchron implementiert werden können. Verwenden Sie bei der Implementierung von Netzwerkaufrufen einen asynchronen, blockierungsfreien Ansatz. Darüber hinaus sollte Ihr Spiel-Backend auch so konfiguriert sein, dass es Ihre Arbeit effizient erledigt, indem Aufgaben in Warteschlangen ausgelagert und schnelle Antworten an Kunden nach Möglichkeit priorisiert werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Beispielsweise kann die Aktualisierung einer Bestenliste am Ende einer Spielsitzung asynchron implementiert werden, sodass der Client nicht warten muss, bis die Aktualisierung der Bestenliste abgeschlossen ist. Implementieren Sie dies stattdessen asynchron auf dem Spielclient und erwägen Sie, Ihren Backend-Service so zu gestalten, dass er diese Art von Vorgängen in Warteschlangen verschiebt, z. B. Amazon SQS. Bei dieser Architektur sollten Sie Ihr Backend so konfigurieren, dass es die Anfrage akzeptiert, sie in SQS in die Warteschlange stellt, wodurch Nachrichten für die asynchrone Verarbeitung dauerhaft gespeichert werden können, und dem Client umgehend antworten. Wenn die Aktualisierung der Bestenliste abgeschlossen ist, kann das Backend ein Update an den Spielclient senden, sodass die Ansicht des Spielers über die Bestenliste aktualisiert wird.

Alternativ kann der Spieler einfach den Bestenlistenbildschirm Ihres Spiels aufrufen, um die neuesten Daten abzurufen, wodurch eine Webanfrage an Ihr Backend gesendet werden kann, um die neuesten Daten aus dem Cache abzurufen.

Implementierungsschritte

- Ermitteln Sie, ob Interaktionen zwischen Client und Backend synchrone Kommunikation erfordern. Implementieren Sie nach Möglichkeit asynchrone, blockierungsfreie Ansätze, um die Ressourcennutzung zu optimieren.
- Verwenden Sie Amazon SQS, um unkritische Aufgaben wie Leaderboard-Updates auszulagern.
- Erlauben Sie dem Client, aktualisierte Daten asynchron abzurufen, z. B. das Abrufen der neuesten Bestenlistendaten bei Bedarf oder über Hintergrundaktualisierungen.

Ressourcen

- [Grundlegendes zu asynchronem Messaging für Microservices](#)
- [Lambda — Verwenden von Serviceintegrationen und asynchroner Verarbeitung](#)

Datenverwaltung

GAMEPERF06: Wie verwaltet und analysiert man Spielserverprotokolle effizient und speichert verschiedene Arten von Spieldaten, um eine optimale Leistung zu erzielen?

Spiele können Spielerdaten, Spielprotokolle und Serverprotokolle enthalten, die nach Möglichkeit voneinander entkoppelt werden sollten. Die Zentralisierung der Protokollaufnahme und des Lebenszyklusmanagements kann Ihren Spielteams zugute kommen, da sie Einblicke in die Geschehnisse im Spiel und auf dem Server bietet.

Best Practices

- [GAMEPERF06-BP01 Zentralisieren Sie die Erfassung und Speicherung von Protokollen](#)
- [GAMEPERF06-BP02 Kategorisieren und speichern Sie Spieldaten basierend auf Zugriffsmustern](#)
- [GAMEPERF06-BP03 Ermöglicht effizientes Formatieren und Batching von Protokollen](#)
- [GAMEPERF06-BP04 Implementieren Sie Richtlinien für die Protokollrotation und Aufbewahrung](#)
- [GAMEPERF06-BP05 Verwenden Sie Überwachungs- und Visualisierungstools](#)

GAMEPERF06-BP01 Zentralisieren Sie die Erfassung und Speicherung von Protokollen

Implementieren Sie eine zentralisierte Lösung für die Erfassung und Speicherung von Protokollen, um Protokolle von Spieleserverinstanzen zu sammeln und GameLift.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Verwenden Sie Dienste wie Amazon CloudWatch Logs, um Protokolldaten von Ihren Spieleservern und GameLift Instances zu sammeln, zu überwachen und zu speichern. CloudWatch Logs bietet eine skalierbare und vollständig verwaltete Lösung für die Protokollverwaltung, die das effiziente Speichern und Abrufen von Protokolldaten ermöglicht, ohne die Leistung des Spieleservers zu beeinträchtigen. Wenn Sie den [CloudWatch Logs-Agenten](#) ausführen, sollten Sie die verschiedenen Installationstypen und Konfigurationsoptionen wie Batchgröße und Pufferdauer berücksichtigen, um die Auswirkungen auf den Spieleserver so gering wie möglich zu halten. Betrachte die Spieleserverinstanzen als kurzlebig und reduziere nach Möglichkeit die Abhängigkeit von lokalisierter Protokollierung. Richten Sie eine zentrale Richtlinie für die Implementierung der bewährten Methoden zur [Protokollierung](#) ein.

Implementierungsschritte

- Verwenden Sie Amazon CloudWatch Logs, um Protokolldaten von Spielservers-Instances zu sammeln, zu überwachen und zu speichern und GameLift so ein zentrales und skalierbares Protokollmanagement zu ermöglichen.

GAMEPERF06-BP02 Kategorisieren und speichern Sie Spieldaten basierend auf Zugriffsmustern

Kategorisieren Sie Ihre Spieldaten anhand ihrer Zugriffsmuster und Speicheranforderungen in verschiedene Typen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Zu den gängigen Kategorien gehören Spielerdaten, Spielstände, persistenter Weltspeicher und Analysedaten.

Implementierungsschritte

Verwenden Sie für jeden Datentyp geeignete Speicherlösungen, um Leistung und Kosteneffizienz zu optimieren:

- **Spielerdaten:** Verwenden Sie Amazon DynamoDB, eine schnelle und skalierbare NoSQL-Datenbank, um Spielerprofile, Einstellungen und Fortschrittsdaten zu speichern. Der Zugriff mit niedriger Latenz und die automatische Skalierung von DynamoDB ermöglichen ein effizientes Abrufen und Aktualisieren von Spielerdaten.
- **Spielstände:** Verwenden Sie Amazon S3, um Spielstände und Checkpoints zu speichern. S3 bietet eine hohe Haltbarkeit und Skalierbarkeit für die Speicherung großer Mengen an Spielständen. Erwägen Sie, S3 Transfer Acceleration oder Amazon CloudFront für schnellere Uploads und Downloads von Spielständen zu verwenden.
- **Persistenter Weltspeicher:** Für Spiele mit persistenten Weltzuständen oder gemeinsam genutzten Spieldaten sollten Sie Amazon DynamoDB, Amazon ElastiCache oder Amazon MemoryDB verwenden. ElastiCache und MemoryDB bieten einen speicherinternen Schlüsselwertspeicher, während DynamoDB eine SSD-gestützte NoSQL-Datenbank ist. Diese Dienste bieten schnellen Zugriff auf gespeicherte Daten und reduzieren so die Zeit, die der Spielserversprozess benötigt, um den Spielstatus zu speichern, was die Gesamtleistung des Prozesses verbessert.

- **Analysedaten:** Verwenden Sie Amazon Managed Streaming for Apache Kafka oder Kinesis Data Streams, um Datenstreams von Ihren Spieledatenproduzenten aufzunehmen. Amazon Managed Service für Apache Flink kann für die Transformation und Analyse in Echtzeit verwendet und zur Verarbeitung und Lieferung an Backend-Data Lakes, Warehouses und Analysedienste an Amazon Data Firehose gesendet werden. In der [Anleitung für Game Analytics Pipeline AWS](#) wird veranschaulicht, wie die Services zusammenarbeiten, um Analysen nahezu in Echtzeit und Batch-Analysen bereitzustellen.

GAMEPERF06-BP03 Ermöglicht effizientes Formatieren und Batching von Protokollen

Konfiguriere deine Gameserver-Prozesse so, dass sie Logs in einem strukturierten und analysierbaren Format wie JSON generieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Implementieren Sie Protokoll-Batching-Techniken, um die Häufigkeit von Protokoll Datenübertragungen von Ihren Spieleservern zum zentralen Protokollspeicher zu minimieren. Durch die Batchverarbeitung von Protokollen wird der Netzwerkaufwand reduziert und die Leistung des Spieleservers verbessert. Verwende ausführliche Logs oder Logs auf Debug-Ebene als Ausnahme und nicht als Standard, da sie zu Leistungs- und Kosteneinbußen führen können, die nach Möglichkeit vermieden werden sollten.

GAMEPERF06-BP04 Implementieren Sie Richtlinien für die Protokollrotation und Aufbewahrung

Richten Sie Richtlinien für die Rotation und Aufbewahrung von Protokollen ein, um das Wachstum der Protokoll Daten zu bewältigen und die Speichernutzung zu optimieren.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Niedrig

Implementierungsleitfaden

Konfiguriere deine Spieleserver so, dass Logs je nach Größe oder Zeitintervallen automatisch rotiert werden. Definieren Sie Richtlinien zur Aufbewahrung von CloudWatch Protokollen in Amazon Logs,

um ältere Protokolldaten, die nicht mehr für aktive Analysen oder Problembehebungen benötigt werden, automatisch zu archivieren oder zu löschen.

GAMEPERF06-BP05 Verwenden Sie Überwachungs- und Visualisierungstools

Verwenden Sie Überwachungs- und Visualisierungstools, um Einblicke in die Leistung Ihres Spieleservers zu gewinnen und Optimierungsmöglichkeiten zu identifizieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Verwenden Sie Amazon CloudWatch , um wichtige Kennzahlen zu überwachen und Alarme für proaktive Benachrichtigungen einzurichten. Nutzen Sie Tools wie Amazon Managed Service for Prometheus und Amazon Managed Grafana, um Metriken von Ihren Spieleservern und Ihrer Infrastruktur zu sammeln, abzufragen und zu visualisieren. Erstellen Sie informative Dashboards, um die Leistung zu verfolgen, Engpässe zu identifizieren und datengestützte Optimierungen vorzunehmen.

Netzwerk und Bereitstellung von Inhalten

GAMEPERF07: Wie gestaltet ihr euren Matchmaking-Service, um die Leistung zu optimieren?

Die Fähigkeiten der Spieler, die Qualität des Internetdienstanbieters (ISP) und die Verteilung der Spielerpopulation müssen bei der Leistungsoptimierung berücksichtigt werden. Spielsitzungen können auf strategisch günstig gelegenen Servern stattfinden, um gleiche Wettbewerbsbedingungen zu schaffen und ein faires Spiel zu veranstalten.

Best Practices

- [GAMEPERF07-BP01 Definieren Sie Schwellenwerte für die Netzwerklatenz für Ihr Spiel](#)
- [GAMEPERF07-BP02 Führe für jeden Spielmodus und jede Spielhosting-Region einen separaten Matchmaking-Dienst aus](#)
- [GAMEPERF07-BP03 Überwachen Sie regelmäßig die Matchmaking-Leistung](#)

- [GAMEPERF07-BP04 Überwachen Sie regelmäßig die Netzwerkleistung](#)
- [GAMEPERF07-BP05 Verwenden Sie Netzwerkbeschleunigungstechnologie, um die Leistung im Internet zu verbessern](#)

GAMEPERF07-BP01 Definieren Sie Schwellenwerte für die Netzwerklatenz für Ihr Spiel

Stellen Sie bei der Entwicklung eines Multiplayer-Spiels sicher, dass Ihre Spielinfrastruktur nicht zu unnötiger Latenz für die Spieler führt. Wenn Ihr Spiel empfindlich auf Netzwerklatenz reagiert, sollten Sie in Ihrer Spielerzuweisungslogik Latenzgrenzwerte festlegen, um Spielern Priorität einzuräumen, die an Spielserverstandorten in der Nähe gehostet werden oder AWS-Regionen die Ihrem Ziel für ein optimales Spielerlebnis entsprechen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

In vielen latenzsensitiven Spielen ist es üblich, die Spielclients so zu instrumentieren, dass sie jeden der Infrastrukturstandorte des Spiels anpingen, um Leistungsdaten wie Netzwerklatenz, Jitter und Paketverlust zu sammeln und diese Daten an das Backend zur Metrikerfassung zu melden, damit sie analysiert werden können. Wenn Sie Spieler zu Spielsitzungen zuordnen, können Sie Ihr Spiel so konfigurieren, dass die vom Spielclient wahrgenommene Netzwerklatenz als eine der Eingaben in Ihre Spielervermittlungsdienst-Logik integriert wird.

GAMEPERF07-BP02 Führe für jeden Spielmodus und jede Spielhosting-Region einen separaten Matchmaking-Dienst aus

Wenn dein Spiel mehrere Spielmodi bietet, aus denen die Spieler wählen können, solltest du die Matchmaking-Systeme für jeden dieser Modi trennen, sodass du die Leistung für jeden Spielmodus unabhängig von den jeweiligen Anforderungen anpassen und Ressourcenkonflikte reduzieren kannst. Für jeden Spielmodus gelten wahrscheinlich eigene Anforderungen in Bezug auf akzeptable Latenz, Match-Größe und andere spielspezifische Matchmaking-Logiken. Sie werden wahrscheinlich auch verschiedene Spielertypen anziehen. Führen Sie den Matchmaking-Dienst der einzelnen Spielmodi als separate Softwarebereitstellung aus, sodass Sie die Leistung der Spielmodi unabhängig voneinander testen und bedienen können.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Sie können diese beispielsweise als separate Lambda-Funktionen für jeden Spielmodus ausführen, oder Sie können sie als separate containerbasierte Servicebereitstellungen betreiben.

Stellen Sie Ihre Matchmaking-Dienste in mehreren Regionen in der Nähe Ihrer Spielserverstandorte bereit. Der Spielerverkehr wird über viele Wege erfolgen, daher ist es wichtig, dass der Matchmaking-Dienst ein up-to-date Latenzprofil über mehrere Kanäle hinweg beibehält, ISPs um die Effizienz der Platzierung von Spielsitzungen mit niedriger Latenz zu verbessern. GameLift FlexMatch bietet zusätzliche Hinweise zur Auswahl von Regionen für Matchmaker und beinhaltet die Möglichkeit, deine Matchmaker in Warteschlangen für Spielsitzungen mit [mehreren Regionen](#) zu integrieren.

GAMEPERF07-BP03 Überwachen Sie regelmäßig die Matchmaking-Leistung

Eine der auffallendsten Möglichkeiten, die Leistung eines Spiels für Spieler zu optimieren, besteht darin, die Wartezeit zu verkürzen, bevor sie an einer Spielsitzung teilnehmen können. Lange Wartezeiten können dazu führen, dass Spieler das Interesse verlieren und zu Fluktuation führen. Daher ist es wichtig, dies bei der Entwicklung Ihrer Matchmaking-Lösung zu berücksichtigen.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Wenn Sie Ihre Matchmaking-Konfiguration für Ihr Spiel entwerfen, sollten Sie Regeln erstellen, die die Bedingungen festlegen, die für die Bildung eines Spiels gelten. Sie sollten die Auswirkungen berücksichtigen, die diese Regeln auf die Leistung des Systems haben werden, insbesondere auf die Wartezeiten für Spieler. Bevor Sie Änderungen an Ihrer Matchmaking-Implementierung vornehmen, wie z. B. das Hinzufügen neuer Bedingungen oder Filter für die Spielerzuweisung, sollten Sie dies vorher testen oder erwägen, diese Änderung schrittweise einer kleinen Gruppe von Spielern als Frühwarnsystem zugänglich zu machen oder A/B Leistungskennzahlen zu sammeln, bevor Sie diese Änderung für die gesamte Spielerpopulation einführen.

Konfigurieren Sie Ihren Matchmaking-Dienst so, dass er detaillierte Protokolle generiert, um die Bedingungen oder Regeln zu verstehen, die auf jede Matchmaking-Anfrage angewendet wurden. Dies hilft bei der Überprüfung und bei der Anpassung der Matchmaking-Implementierung nach Bedarf.

[Amazon GameLift FlexMatch](#) bietet beispielsweise einen vollständig verwalteten Matchmaking-Service an, der als eigenständiger Dienst mit Ihrem eigenen Spielserver-Hosting oder mit

auf Amazon GameLift gehosteten Spieleservern verwendet werden kann. FlexMatch kann Event-Benachrichtigungen an Amazon generieren EventBridge, siehe [FlexMatch Event-Benachrichtigungen einrichten](#). Verwenden Sie Amazon Simple Notification Service (Amazon SNS), um Matchmaking-Daten im JSON-Format zu empfangen, sodass Sie diese Informationen automatisch verarbeiten und speichern können, um die Matchmaking-Leistung zu verbessern.

Richten Sie Messwerte ein, um zu verfolgen, wie lange Ihr Matchmaking-Service benötigt, um eine passende Spielsitzung für Spieler zu finden. Überprüfe regelmäßig die Messwerte für die Dauer der Spielerzuweisung und korreliere diese Zeiten mit dem Spielerverhalten und der Stimmung in der Community. Verwenden Sie diese Daten, um geeignete Schwellenwerte für Matchmaking-Timeouts zu entwickeln, die in Ihre Konfiguration der Matchmaking-Regeln aufgenommen werden können.

Amazon GameLift FlexMatch bietet beispielsweise Unterstützung bei der Definition von Timeouts für Matchmaking-Anfragen sowie bei der Erstellung von Matchmaking-Regeln, die [es ermöglichen, die Anforderungen im Laufe der Zeit zu lockern](#). Mit dieser Funktion können Sie Matchmaking einrichten, das sich anpassen lässt, sodass es einfach ist, Matches zu erstellen und Spieler in Spielsitzungen unterzubringen, wenn Matches schwer zu finden sind.

GAMEPERF07-BP04 Überwachen Sie regelmäßig die Netzwerkleistung

Für kompetitive Spiele ist es wichtig, ein einheitliches Spielerlebnis zu bieten.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Ein Spiel, das für eine größere Spielerbasis zuverlässig 50 ms dauert, ist fairer und unterhaltsamer als ein Spiel, bei dem ein Spieler 10 ms Ping und ein anderer 70 ms Ping hat. Änderungen des ISP-Routings können sich auf einen Teil der Spielerpopulation auswirken, und dein Matchmaking-System muss sich anpassen. [Amazon CloudWatch Network Monitoring](#) hilft Ihnen dabei, festzustellen, ob das Problem bei Ihrem Spiel oder beim Internetanbieter des Spielers liegt.

Implementierungsschritte

- Verwenden Sie Amazon Cloudwatch Network Monitoring, um die Netzwerkleistung zu verfolgen und Routing-Probleme zu identifizieren.
- Verwenden Sie VPC Flow Logs, um ungewöhnliche Verkehrsmuster oder verworfene Pakete zu identifizieren, die auf Netzwerküberlastungen, ISP-Probleme oder Fehlkonfigurationen hinweisen können, die sich auf die Spielerlatenz auswirken.

GAMEPERF07-BP05 Verwenden Sie Netzwerkbeschleunigungstechnologie, um die Leistung im Internet zu verbessern

Sie können die latenzempfindliche Spielinfrastruktur nicht nur physisch näher an den Spielern platzieren, sondern auch das Spielerlebnis verbessern, indem Sie die Netzwerkleistung für Ihr Spiel optimieren. AWS verwendet das BGP-Protokoll, um das [Internet-Routing](#) so zu beeinflussen, dass der schnellste Weg von Internetdiensteanbietern zu unserem Grenznetz genutzt wird. Wenn du dein eigenes Netzwerk betreibst und mehr Kontrolle und Beobachtbarkeit über das Routing-Verhalten und die BGP-Werbung benötigst, kannst du privates [Peering](#) verwenden oder den Verkehr vom Internet Direct Connect zu deinem Spiel weiterleiten, auf dem es läuft. AWS

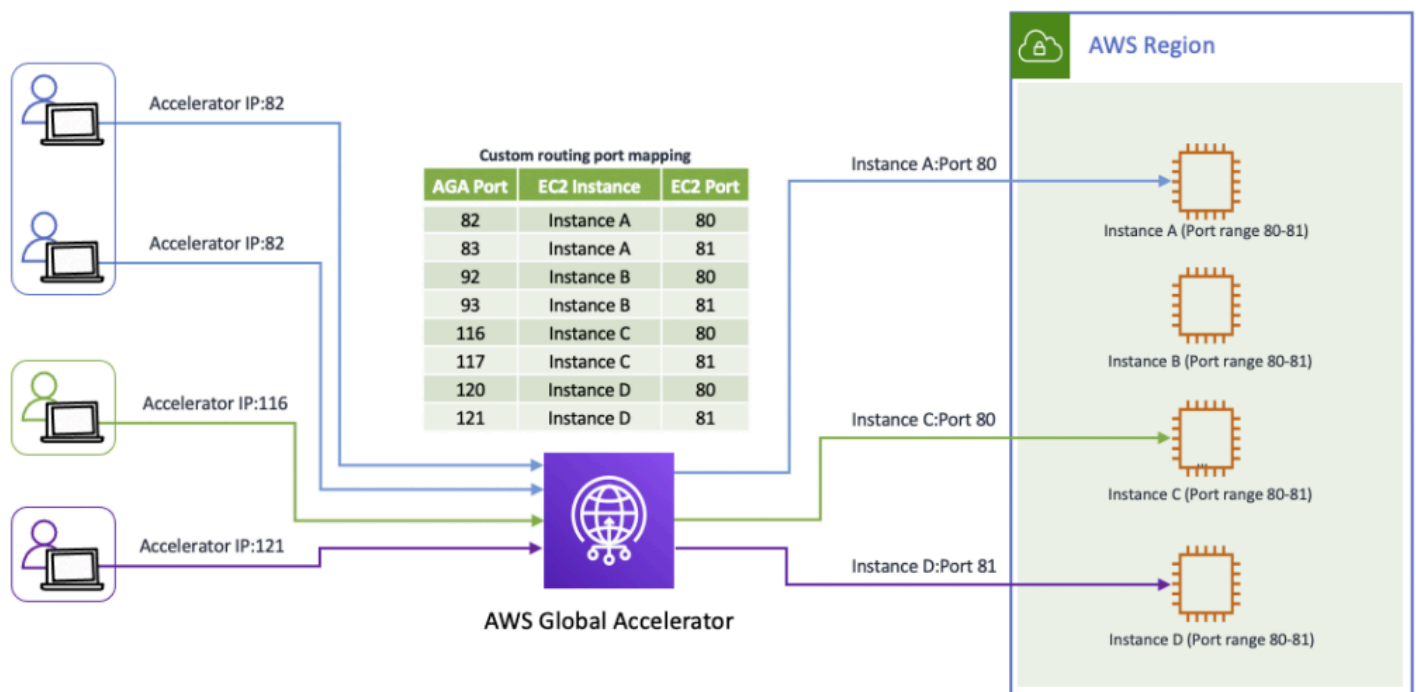
Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Ziehen Sie die folgende Referenzarchitektur in Betracht, um die Leistung und Reaktionsfähigkeit des Internets zu verbessern.

Verbesserte Netzwerkleistung für Spiele mit Global Accelerator

Als vollständig verwaltete Lösung für Netzwerk-Routing verbessert [AWS Global Accelerator](#) die Netzwerkleistung Ihrer Anwendung mithilfe des AWS globalen Netzwerks. Dies kann verwendet werden, um Ihren Gameplay-Verkehr, Voice-Chat und Echtzeit-Messaging-Verkehr sowie andere latenzempfindliche Anwendungen zu beschleunigen und gleichzeitig einen schnellen Failover für Ihre Spieleserver zu ermöglichen. Die [benutzerdefinierten Routing-Beschleuniger](#) von Global Accelerator können in Ihren Matchmaking-Dienst integriert werden, um mithilfe statischer Anycast-IP-Adressen und -Ports ein deterministisches Routing mehrerer Spieler zu derselben Spielsitzung zu ermöglichen.



Ihre Spieleentwicklungsteams sind möglicherweise auf der ganzen Welt verteilt und benötigen leistungsstarken Zugriff auf gemeinsam genutzte Inhalte oder Ressourcen. Um die Leistung für gemeinsam genutzte Inhalte zu verbessern, die in Amazon S3-Buckets gespeichert sind, können Sie mithilfe von [S3 Cross-Region Replication](#) eine bidirektionale Replikation Ihrer Daten über Regionen hinweg einrichten, sodass Benutzer auf Daten aus Buckets zugreifen können, die sich näher an ihnen befinden. Um dieses Zugriffsmuster zu vereinfachen, verwenden Sie [S3 Multi Region Access Points](#), die Anfragen an S3 über das globale Netzwerk mithilfe von Global Accelerator beschleunigen.

Weitere Informationen finden Sie unter [Verbesserung des Spielerlebnisses durch die Nutzung von AWS Global Accelerator und Amazon GameLift FleetIQ](#).

Implementierungsschritte

- Verwenden Sie AWS Global Accelerator, um die Netzwerkleistung für Gameplay-Verkehr, Voice-Chat und Echtzeit-Messaging zu verbessern und gleichzeitig einen schnellen Failover zu Spielsevernen zu ermöglichen.
- Konfigurieren Sie die benutzerdefinierten Routing-Beschleuniger von Global Accelerator so, dass sie in Ihren Matchmaking-Service integriert werden können und so mithilfe von statischem Anycast eine deterministische Weiterleitung von Spielern zu Spielsitzungen ermöglichen. IPs
- Aktivieren Sie die regionsübergreifende S3-Replikation, um gemeinsam genutzte Inhalte regionsübergreifend für Spieleentwicklungsteams zu replizieren.

- Verwenden Sie S3-Zugriffspunkte für mehrere Regionen, um den S3-Datenzugriff über das AWS globale Netzwerk für weltweit verteilte Benutzer zu beschleunigen.

Prozess und Kultur

GAMEPERF08: Wie stimmen Sie die Performance-Messlatte für Ihr Spiel mit den Erwartungen der Spieler und Entwickler ab?

Die Kenntnis Ihres Spielers und Entwicklers ist einer der wichtigsten Aspekte zur Verbesserung der Leistungseffizienz. Die Bereitstellung eines leistungsstarken Spiels mit geringem Betriebskosten ist eine der besten Möglichkeiten, Spielern und Entwicklern zu zeigen, dass Ihnen ihre Erfahrung wichtig ist und Sie Ihr Spiel und Ihr Studio von der Konkurrenz abheben können.

Best Practices

- [GAMEPERF08-BP01 Informiere den Spieler und beziehe ihn in deinen Prozess mit ein](#)
- [GAMEPERF08-BP02 Passen Sie die Lösungsauswahl an den Fähigkeiten und dem Fachwissen des Entwicklungsteams an](#)

GAMEPERF08-BP01 Informiere den Spieler und beziehe ihn in deinen Prozess mit ein

Bieten Sie eine Option zur Anzeige von Metriken wie Latenz, Frames pro Sekunde und verworfene Pakete im Spiel an. Probleme mit der Oberflächeninfrastruktur und Wartungsausfälle durch Kommunikation mit Spielern wie Statusseiten. Feiere neue Spielorte mit Spielerkommunikation, einschließlich Entwickler-Blogs, und setze Erwartungen an erwartete Verbesserungen des Spielerlebnisses.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Schließe den Spieler mit ein

Bieten Sie einen einfachen Prozess zur Einreichung von Diagnosen an, bei dem relevante Dateien gesammelt und an ein Support-Ticket für Spieler von Ihrem Spielclient angehängt

werden. Ermöglichen Sie ein Support-Forum, in dem sich Spieler gegenseitig helfen und Teil der Verbesserung des Spielerlebnisses werden können

Ziehe Kompromisse und Spielererwartungen in Betracht

Die Umstellung von Backend-Systemen aus Gründen der Kosteneffizienz mag für Spieler nicht auffallen, aber die Umstellung von Spieleservern kann die Ping-Zeit verändern. Seien Sie konsistent und fair gegenüber den Spielern, wenn Sie die Gründe für die Erweiterung und Reduzierung Ihrer Spiele-Hosting-Standorte angeben.

Spielergemeinschaften und Regionen haben ihre eigenen Merkmale, die sich auf die Erwartungen an Ihr Spiel auswirken können. Südkorea hat zum Beispiel eines der schnellsten Internetverbindungen der Welt, und die Erwartung an das Gameplay ist eine einstellige Latenz, was zu einem hart umkämpften Spiel führt. Gelegenheitsspiel auf Mobilgeräten führt zu einem anderen Leistungsprofil und Nutzungsmuster als beim Spielen auf Konsolen und PCs.

Anmeldung und Lobby sind Teil des Spielerlebnisses und sollten sich anfühlen, selbst wenn der Server wegen Wartungsarbeiten offline ist. Die Planung von Schlachtzügen oder das Herumhängen in der Lobby sind Teil des Spielerlebnisses und sollten bei der Auswahl von Schwerpunkten aus Gründen der Leistungseffizienz unbedingt berücksichtigt werden. Spieler können deinen Spielclient monatelang geöffnet lassen, manchmal melden sie sich einfach gelegentlich an, um die Patchnotizen zu lesen. Bei einem Live Ops-Spiel muss das gesamte Spielererlebnis als Teil des Entwicklungsprozesses und der Kultur berücksichtigt werden.

Implementierungsschritte

- Stellen Sie spielinterne Messwerte wie Latenz, FPS und Paketverlust bereit und kommunizieren Sie Infrastrukturprobleme und Wartungspläne über Statusseiten und Updates für Spieler.
- Implementiere im Spielclient eine Funktion zum Abspeichern und Einreichen von Diagnosen und richte ein Support-Forum ein, um die von der Community betriebene Problembehebung und Verbesserung zu fördern.
- Passen Sie die Leistungsoptimierungen an die Erwartungen der Spieler-Community an, z. B. geringe Latenz für wettbewerbsfähige Regionen oder ein responsives login/lobby Spielerlebnis für Gelegenheitsspieler und Spieler mit langen Sessions.
- Entwerfen Sie Live-Ops-Workflows so, dass sie das gesamte Spielererlebnis berücksichtigen, vom aktiven Gameplay bis hin zum Verhalten ungenutzter Kunden, und so eine reibungslose Interaktion ermöglichen.

GAMEPERF08-BP02 Passen Sie die Lösungsauswahl an den Fähigkeiten und dem Fachwissen des Entwicklungsteams an

Beurteilen Sie bei der Auswahl Ihrer Hosting-Option die Fähigkeiten und das Fachwissen Ihres Teams bei der Verwaltung und Optimierung der Leistung von Spieleservern. Selbst gehostete Lösungen wie EC2 Container erfordern mehr Wissen über Infrastrukturmanagement, Leistungsoptimierung und Skalierung. Wenn Ihrem Team diese Fähigkeiten fehlen, ist ein Managed Service wie dieser GameLift möglicherweise besser geeignet, da er viele der Komplexitäten abstrahiert und es Ihrem Team ermöglicht, sich auf spielspezifische Optimierungen zu konzentrieren.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Indem Sie diese Faktoren bewerten und Leistungstests für verschiedene Hosting-Optionen durchführen, können Sie die am besten geeignete Lösung auswählen, die den spezifischen Anforderungen Ihres Spiels entspricht und gleichzeitig die Leistungseffizienz optimiert.

Ressourcen

Erfahre mehr über unsere Best Practices im Bereich Leistungseffizienz.

Zugehörige Dokumente:

- [AWS -Architekturzentrum](#)
- [Säule der Leistungseffizienz — AWS Well-Architected Framework](#)
- [Vergleich Ihrer lokalen Speichermuster mit Speicherdiensten AWS](#)
- [Instanzspeicher, temporärer Blockspeicher für EC2 Instanzen](#)
- [Erfassen Sie mithilfe des CloudWatch Agenten Metriken, Logs und Traces](#)
- [CloudWatchAgent](#)
- [Wie aktiviere und konfiguriere ich Enhanced Networking auf meinen EC2 Instances?](#)
- [Verbesserung des Spielerlebnisses durch die Nutzung von AWS Global Accelerator und Amazon FleetIQ GameLift](#)
- [Technologie-Blog von Riot Games: Skalierbarkeit und Belastungstests für Valorant](#)
- [Hyperskalierbare Online-Spiele mit einer Hybridlösung AWS](#)
- [USE-Methode \(Usage Saturation and Errors\)](#)

- [Amazon EC2 Spot-Instances](#)
- [Skalierbare Leistung mit Amazon ElastiCache](#)
- [Strategien zum Datenbank-Caching mit Redis](#)
- [Amazon Virtual Private Cloud Connectivity Options](#)
- [Bewährte Entwurfsmuster: Optimierung der Amazon S3 S3-Leistung](#)

Verwandte Benchmarks:

- [Amazon EBS-Volumen vergleichen](#)
- [Netzwerkbandbreite der Amazon EC2 Instance](#)

Zugehörige Tools:

- [Unreal Engine: Testen und Optimieren Ihrer Inhalte](#)
- [Unity Profiler](#)
- [Öffnen Sie das Qualitätssystem 3D Engine \(O3DE\)](#)
- [Überwachung von GameLift Amazon-Servern](#)
- [GameLift Test-Toolkit von Amazon](#)

Zugehörige Videos:

- [AWS re:Invent 2019: \[WIEDERHOLE 2\] EC2 Amazon-Stiftungen \(CMP211-R2\)](#)
- [AWS re:Invent 2019: Unterstützung für Amazon der nächsten Generation EC2: Tiefer Einblick in das Nitro-System \(03-R2\) CMP3](#)
- [Erste Schritte mit Amazon GameLift FleetIQ — AWS Online Tech Talks](#)
- [Zach Blitz von Riot Games über die Verwendung zur Verbesserung des Spielens AWS](#)
- [AWS re:Invent 2023 — AWS Graviton: Das beste Preis-Leistungs-Verhältnis für Ihre Workloads \(\) AWS CMP334](#)

Zugehörige Schulungen:

- [Gameserver-Hosting auf AWS](#)
- [Amazon GameLift Fleet IQ für Spieleserver verwenden](#)

- [Erste Schritte mit AWS for Games — Teil I](#)
- [Gameserver-Hosting auf AWS](#)
- [AWS re:Invent 2023 — AWS Graviton: Das beste Preis-Leistungs-Verhältnis für Ihre AWS Workloads \(\) CMP334](#)

Kostenoptimierung

Die Säule der Kostenoptimierung umfasst den kontinuierlichen Prozess der Verfeinerung und Verbesserung eines Systems über seinen gesamten Lebenszyklus. Dieser Prozess reicht vom ersten Entwurf Ihres ersten Machbarkeitsnachweises bis hin zum laufenden Betrieb der Produktionsworkloads. Wenn Sie die in diesem paper beschriebenen Verfahren anwenden, können Sie kostenbewusste Systeme entwickeln und betreiben, mit denen Sie die gewünschten Geschäftsergebnisse zu einem niedrigsten Preis erzielen. Durch die Implementierung dieser Methoden zur Kostenoptimierung kann Ihr Unternehmen den Wert Ihrer Cloud-Investition maximieren.

Spiele sind einzigartige kreative Projekte, die um die Aufmerksamkeit und Spielzeit der Spieler konkurrieren müssen. Vor der Veröffentlichung haben Spieleentwickler oft keine klare Vorstellung davon, wie beliebt oder langlebig ihr Spiel werden wird. Abhängig von der Monetarisierungsstrategie, den Geschäftsprioritäten und der Lebenszyklusphase des Spiels müssen Entwickler bei der Bewertung von Entscheidungen zur Kostenoptimierung Kompromisse eingehen.

In der Phase vor der Veröffentlichung eines mit Spannung erwarteten neuen Spiels liegt der Fokus beispielsweise in der Regel auf der Entwicklung von Funktionen und der speed-to-market Leistung. Die Priorität besteht darin, zu überprüfen, ob die Infrastruktur skalierbar ist, um der hohen Nachfrage der Spieler gerecht zu werden. Umgekehrt, wenn ein Spiel nicht erfolgreich ist oder sich die Entwicklung verlangsamt, kann sich der Schwerpunkt darauf verlagern, die Kosten so weit wie möglich zu senken, um das Spiel weiterhin für bestehende Spieler zu betreiben.

Viele Spieleentwickler betreiben auch mehrere Spiele gleichzeitig, was zusätzliche Überlegungen erfordert. Ressourcen wie Infrastruktur, Software und Mitarbeiter können für mehrere Live-Spiele gemeinsam genutzt werden, sodass Verluste aus einem Spiel durch Gewinne aus einem anderen ausgeglichen werden können. In diesem Szenario kann ein Fokus auf die Kostenoptimierung die Finanzen des gesamten Spieleportfolios verbessern.

Angesichts der einzigartigen Geschäftsmodelle, des Umfangs und der Unvorhersehbarkeit von Spielen können die folgenden Schlüsselfragen als Grundlage für Entscheidungen zur Kostenoptimierung dienen:

- Wie kann ich die Infrastrukturkosten pro Spieler, System und Spielfunktion messen?
- Was ist das richtige Gleichgewicht zwischen Kostenoptimierung und Spielererlebnis für die aktuelle Lebensphase meines Spiels?

- Wie kann ich die Kapitalrendite maximieren, indem ich das richtige Preismodell für meine AWS Ressourcen verwende?

Die Anwendung dieser bewährten Methoden und das Stellen der richtigen Fragen können Spieleentwicklern helfen, kostenbewusste Systeme zu entwickeln und zu betreiben, mit denen Geschäftsergebnisse erzielt und gleichzeitig die Kosten minimiert werden.

Schwerpunktbereiche

- [Designprinzipien](#)
- [Praxis für Cloud-Finanzmanagement](#)
- [Ausgabenerkennung und Nutzungsbewusstsein](#)
- [Kostengünstige Ressourcen](#)
- [Datenübertragungskosten](#)
- [Verwaltung von Nachfrage- und Angebotsressourcen](#)
- [Optimierung im Laufe der Zeit](#)
- [Ressourcen](#)

Designprinzipien

Zusätzlich zu den Designprinzipien aus dem Bereich Kostenoptimierung des Well-Architected Framework optimieren die folgenden Entwurfsprinzipien die Kosten für den Betrieb Ihres Spiel-Workloads in der Cloud.

- Messen Sie die Infrastrukturkosten pro Spieler, System und Spielfunktion: Verschaffen Sie sich einen Überblick über die Infrastrukturkosten, die für bestimmte Spielererlebnisse und Funktionen in verschiedenen Spielsystemen erforderlich sind, und verfolgen Sie diese. Auf diese Weise können Bereiche Ihrer Architektur identifiziert werden, in denen möglicherweise eine Kostenoptimierung erforderlich ist.
- Beurteilen Sie den Kompromiss zwischen Kostenoptimierung und Spielererlebnis: Beurteilen Sie, in welcher Phase sich Ihr Spiel befindet, um den richtigen Schwerpunkt zu finden — Spielererlebnis oder Kostenoptimierung. Sobald ein Spiel die kritische Masse erreicht hat und sich die Spielerzahl stabilisiert hat, ist es in der Regel an der Zeit, sich auf die Optimierung der Betriebskosten zu konzentrieren. Der Schlüssel liegt darin, ein ausgewogenes Verhältnis zwischen der Bereitstellung eines großartigen Spielerlebnisses und dem kostengünstigsten Betrieb Ihrer Spieleinfrastruktur zu finden. Die Umsetzung dieser Designprinzipien maximiert die Rendite Ihrer Spielinvestitionen.

Praxis für Cloud-Finanzmanagement

Es gibt keine spezifischen Best Practices für Cloud Financial Management für Games Lens. Hinweise zum Cloud-Finanzmanagement finden Sie in der [Säule Kostenoptimierung des Well-Architected Framework](#).

Ausgabenerkennung und Nutzungsbewusstsein

GAMECOST01: Wie messen Sie die Kosten Ihrer Spielumgebungen?

Verstehe die Kosten pro Spieler, Spielfunktion und Umgebung, sodass du deine Ausgaben verwalten und prognostizieren kannst, wenn sich die Anzahl der Spieler im Laufe der Zeit ändert und Funktionen hinzugefügt und verbessert werden. Beachten Sie die folgenden bewährten Methoden, um die Kosten Ihrer verschiedenen Spielumgebungen im Griff zu behalten.

Best Practices

- [GAMECOST01-BP01 Implementieren Sie die Kostenzuweisung pro Spieler, Spielfunktion und Umgebung](#)
- [GAMECOST01-BP02 Entdecken Sie Optimierungsmöglichkeiten](#)

GAMECOST01-BP01 Implementieren Sie die Kostenzuweisung pro Spieler, Spielfunktion und Umgebung

Die Kostenzuweisung für Spieleserver ist in der Regel einfacher durchzuführen als für Backend-Dienste für Spiele, da ein Spieleserver normalerweise so optimiert ist, dass er eine bestimmte Anzahl gleichzeitiger Spieler pro Instanz hosten kann, die sich über die Kosten für den Betrieb der Instanz amortisieren kann.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Bei Backend-Diensten für Spiele empfiehlt es sich, die Komponenten Ihres Spiels in einzelne Funktionen zu unterteilen, die als separate logische oder physische Ressourcen verwaltet werden können, um die Kostenanalyse zu vereinfachen.

Obwohl es beispielsweise einfach erscheinen mag, eine einzige monolithische Anwendung zum Hosten von Spiel-Backend-Diensten zu implementieren, macht es dieses Muster schwierig, die Gesamtkosten pro Spieler und Spielfunktion im Laufe der Zeit abzuleiten, wenn Sie weitere Funktionen hinzufügen, da die Rechen-, Netzwerk- und Speicherkosten der Ressourcen auf alle Funktionen aufgeteilt werden. Erwägen Sie die Einführung einer serverlosen Architektur für Ihre Spiele-Backend-Services mit Diensten wie [Amazon API Gateway](#) AWS Lambda und/oder AWS Fargate für Datenverarbeitung, [Amazon SQS und Amazon SNS](#) für Messaging, Amazon S3 für Objektspeicher und Amazon DynamoDB für Datenbankspeicher. Diese Services sind nur einige Beispiele für Produkte, deren Preisgestaltung nutzungsbasiert und in erster Linie vom Anforderungsvolumen abhängt, sodass die Kosten detailliert visualisiert werden können. Einzelne Ressourcen wie Lambda-Funktionen, Fargate-Dienste, DynamoDB-Tabellen und S3-Buckets können mit Kostenzuweisungs-Tags verknüpft werden, sodass Sie die Kosten dieser Dienste den Namen der Spielfunktionen zuordnen können, sodass Sie die Kosten für jeden Ihrer Dienste leicht nachvollziehen können.

Es wird außerdem empfohlen, jede Ihrer Spieleentwicklungsumgebungen separat zu verwalten, damit Sie die Kosten für die verschiedenen Umgebungen zuordnen können. In der Regel verwalten Spieleentwickler separate Umgebungen für Entwicklungs-, Test-, Staging- und Produktionsumgebungen, wie im Abschnitt Betrieb dieser Spielebranche beschrieben. Jede Umgebung hat normalerweise unterschiedliche Skalierbarkeits-, Leistungs- und Nutzungsanforderungen und kann von separaten Teams verwaltet werden. Um die Kosten unter Kontrolle zu halten, sollten Sie diese Umgebungen so organisieren, dass Sie die Kosten jeder Umgebung ordnungsgemäß überwachen und zuordnen können.

Weitere Informationen finden Sie in der folgenden Dokumentation:

- [Entwicklung eines serverlosen Mehrspieler-Spiels, das skalierbar ist](#)
- [Eigenständige Spielesitzungsserver mit einem basierten Backend WebSockets](#)
- [Eigenständige Spielesitzungsserver mit einem serverlosen Backend](#)

Implementierungsschritte

- Untergliedern Sie Backend-Services für Spiele mithilfe serverloser oder containerisierter Architekturen wie AWS Lambda Amazon API Gateway in unterschiedliche Funktionen und ermöglichen Sie so eine detaillierte AWS Fargate Kostenzuweisung pro Funktion.

- Wenden Sie Kostenzuweisungs-Tags auf einzelne Ressourcen an (z. B. Lambda-Funktionen, DynamoDB-Tabellen und S3-Buckets), um Kosten bestimmten Spielfunktionen zuzuordnen und so eine bessere Kostenanalyse zu ermöglichen.
- Verwalten Sie separate Umgebungen für Entwicklung, Test, Staging und Produktion und organisieren und überwachen Sie deren Kosten unabhängig voneinander, um sie an die Skalierbarkeit und die Nutzungsanforderungen anzupassen.

GAMECOST01-BP02 Entdecken Sie Optimierungsmöglichkeiten

Spieleentwickler und Herausgeber können AWS FinOps Methoden nutzen, um ihre Cloud-Kosten zu optimieren und sich einen besseren Überblick über ihre Cloud-Ausgaben zu verschaffen. Auf diese Weise können Spielehersteller die durchschnittlichen Kosten, die für die Wartung der Infrastruktur für die Spieler erforderlich sind, mit den finanziellen Ergebnissen des Spiels in Einklang bringen.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Niedrig

Implementierungsleitfaden

AWS bietet eine gebrauchsfertige [Lösungsberatung für Cloud Financial Management](#), mit der Sie Ihre Ausgaben für Cloud-Dienste verwalten und optimieren können. Diese Funktion umfasst detaillierte Transparenz sowie Kosten- und Nutzungsanalysen zur Unterstützung der Entscheidungsfindung bei Themen wie Ausgaben-Dashboards, Optimierung, Ausgabenlimits, Rückbelastungen sowie Erkennung und Reaktion auf Anomalien. Die Lösungsberatung für Cloud Financial Management umfasst Budget- und Prognosefunktionen, die Ihnen eine definierte, kostenoptimierte Architektur für Ihre Workloads bieten, sodass Sie das richtige Preismodell auswählen und die für Ihre Teams relevanten Ressourcenkosten zuordnen können. Dadurch werden Techniken zur Nachverfolgung, Benachrichtigung und Kostenoptimierung in Ihrer gesamten Umgebung und Ihren Ressourcen aktiviert. Sie können Ausgabeninformationen zentral verwalten und wichtigen Stakeholdern bei Bedarf Zugriff gewähren, um einen gezielten Überblick zu erhalten und die Entscheidungsfindung zu unterstützen.

Ein weiteres wichtiges FinOps Tool ist der [Cost Optimization Hub](#), der einen zentralen Überblick über Empfehlungen und Möglichkeiten zur Kostenoptimierung in Ihrer AWS-Konten gesamten Region bietet AWS-Regionen, sodass Sie das Beste aus Ihren AWS Ausgaben herausholen können. Sie können den Cost Optimization Hub verwenden, um Empfehlungen zur AWS Kostenoptimierung in Ihrem Land zu identifizieren, zu filtern AWS-Konten und zu aggregieren AWS-Regionen. Es enthält Empfehlungen zur richtigen Dimensionierung von Ressourcen, zum Löschen inaktiver Ressourcen,

zu Savings Plans und Reserved Instances. Mit einem einzigen Dashboard müssen Sie nicht mehrere AWS Produkte aufrufen, um Möglichkeiten zur Kostenoptimierung zu identifizieren.

Wenn Ihre Spielerteams gemeinsam genutzt werden, können Sie AWS-Konten die [Option „Meine Anwendungen auf der AWS-Managementkonsole Startseite“](#) verwenden, um die Ressourcenkosten der Anwendungen für einzelne Workloads einzusehen. Diese detaillierte Ansicht ermöglicht es Ihnen, die spezifischen Kostentrends innerhalb Ihrer Spieleinfrastruktur zu identifizieren, sodass Sie fundierte Entscheidungen über die Ressourcenzuweisung und -optimierung treffen können.

Darüber hinaus deckt die regelmäßige Überprüfung Ihrer Abrechnungs- und Kostenmanagementdaten mithilfe von [AWS Datenexporten](#) versteckte Möglichkeiten zur Kosteneinsparung auf. Dieser detaillierte Bericht bietet eine umfassende Aufschlüsselung Ihrer Cloud-Ausgaben, sodass Sie Bereiche identifizieren können, in denen Sie zu viel ausgeben, ungenutzte Ressourcen nutzen und Möglichkeiten zur Inanspruchnahme kostengünstigerer Dienste oder Preismodelle erkennen können.

Durch die Übernahme von FinOps Prinzipien und die Nutzung der von AWS bereitgestellten Tools können Spieleentwickler und Herausgeber ihre Cloud-Ressourcen so effizient wie möglich nutzen, was letztlich ihr Geschäftsergebnis steigert und Mittel für die weitere Spieleentwicklung und Innovation freisetzt.

Implementierungsschritte

- Nutzen Sie AWS Cloud Finanzmanagement-Tools für eine detaillierte und detaillierte Transparenz, Ausgaben-Dashboards, die Erkennung von Anomalien und die Kostenzuweisung, um Cloud-Ausgaben effektiv zu optimieren und nachzuverfolgen.
- Verwenden Sie den Cost Optimization Hub, um Rightsizing, Savings Plans und Reserved Instance-Empfehlungen für alle AWS-Konten Regionen zu zentralisieren.
- Überprüfen Sie regelmäßig die AWS Abrechnungsdaten mithilfe von Datenexporten usw. MyApplication AWS , um workloadspezifische Kosten zu analysieren, Einsparmöglichkeiten aufzudecken und die Ressourcenzuweisung zu optimieren.

Kostengünstige Ressourcen

GAMECOST02: Wie wählst du die richtige Rechenlösung für deine Spieleserver aus?

Einer der einzigartigsten Aspekte einer Spielauslastung im Vergleich zu anderen Arten von Workloads ist der Spieleserver. Der Spieleserver ist für das Spielerlebnis von entscheidender Bedeutung, da die Spieler von ihrem Spielclient aus eine Verbindung zu ihm herstellen, um eine Spielsitzung zu spielen.

Der Spieleserver ist auch einer der größten Kostentreiber für den Betrieb eines Multiplayer-Spiels. Daher ist es wichtig, die Art und Weise, wie Sie die Computerinfrastruktur für Ihre Spieleserver nutzen, zu optimieren, um die Kosten zu senken.

Best Practices

- [GAMECOST02-BP01 Optimieren Sie die Kosten für die Datenübertragung über das Internet](#)
- [GAMECOST02-BP02 Optimiere die Anzahl der Spielsitzungen, die auf jeder Spieleserverinstanz gehostet werden, um die Kosten zu optimieren](#)
- [GAMECOST02-BP03 Wählen Sie die entsprechende Option zur Preisberechnung aus, um die Kosten zu senken](#)

GAMECOST02-BP01 Optimieren Sie die Kosten für die Datenübertragung über das Internet

Während AWS in erster Linie Gebühren für ausgehende (ausgehende) Datenübertragungen von Ihren AWS Ressourcen ins Internet anfallen, können Spielefirmen mit hohen Kosten im Zusammenhang mit der Datenübertragung über AWS Direct Connect oder AWS Gateway-Loadbalancer konfrontiert werden, die sowohl für eingehende (eingehende) als auch für ausgehende Daten Gebühren erheben können. Implementieren Sie Lösungen, die die Gesamtkosten für die Übertragung von Daten vom AWS Backend Ihres Spiels zu Ihren Spielern reduzieren. Konzentrieren Sie sich dabei auf die Minimierung der ausgehenden Gebühren für Ihre AWS Ressourcen sowie auf die Bewertung von Optionen zur Verwaltung von Eingangs- und Ausgangsgebühren durch AWS Konnektivitätsdienste.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Verwenden Sie Amazon CloudFront , um die Kosten für die Bereitstellung von Inhalten und für öffentlich zugängliche Webanwendungen zu senken.

Spielinhalte und -ressourcen, die in der Cloud gespeichert sind, werden in der Regel in Amazon S3 gespeichert und entweder direkt von S3 oder von in Amazon gehosteten Webservern EC2 ,

die die Inhalte von Amazon S3 abrufen und an Kunden liefern, an den Spieleclient übermittelt. Um die Datenübertragungskosten beim Herunterladen von Inhalten zu senken, sollten Sie Amazon CloudFront vor Ihrem Cloud-Speicher verwenden, um Inhalte für Benutzer bereitzustellen.

Durch die Verwendung CloudFront können die Kosten für die Datenübertragung gesenkt werden, da die Bereitstellung Ihrer Inhalte aus Regionen günstiger ist CloudFront points-of-presence als die direkte Bereitstellung von Inhalten aus Regionen, und es fallen CloudFront keine Gebühren für AWS den Abruf von Ursprungsländern an, z. B. Amazon EC2 und Amazon S3. Wenn Ihre Inhalte statisch sind und sich nicht häufig ändern, können CloudFront Sie diese Daten in der Nähe der Endnutzer zwischenspeichern, wodurch die Kosten weiter gesenkt werden können.

CloudFront verbessert auch die Kosteneffizienz von öffentlich zugänglichen Webanwendungen und -diensten an der Vorderseite, auch wenn kein Caching verwendet wird, da die Kosten für die Datenübertragung zwischen Ihren Servern und Clients durch die Weiterleitung des Datenverkehrs über das Netzwerk gesenkt werden können. AWS

[Amazon CloudWatch](#) kann verwendet werden, um Ihre CloudFront Amazon-Nutzung zu überwachen. Für Anwendungsfälle, in denen Sie mehrere Content Delivery Networks (CDNs) verwenden, kann [Amazon CloudFront Origin Shield](#) eine zusätzliche Caching-Ebene bereitstellen, um die Anzahl der ursprünglichen Anfragen von verschiedenen Anbietern zu konsolidieren und zu reduzieren.

Um den Netzwerkverkehr Ihres Spiels zu verstehen, können Sie [VPC Flow Logs](#) und [Amazon CloudWatch Internet Monitor](#) aktivieren, um end-to-end Einblick in die Backend-Verbindungen von Spielern oder Spielen zu erhalten. Mit diesem Ansatz können die Ursachen für hohe Datenübertragungskosten ermittelt und architektonische Änderungen vorgenommen werden, um die Ausgaben für die Datenübertragung zu optimieren.

Implementierungsschritte

- Verwenden Sie Amazon CloudFront vor Amazon S3 oder EC2 basierten Inhaltsquellen, um die Datenübertragungskosten zu senken, indem Sie kostengünstigere Versandgebühren nutzen CloudFront points-of-presence und die Gebühren für den Abruf von Ursprungsländern abschaffen.
- Aktivieren Sie VPC Flow Logs und Amazon CloudWatch Internet Monitor, um den Netzwerkverkehr zu analysieren und architektonische Änderungen zu identifizieren, um die Datenübertragungskosten zu optimieren.
- Implementieren Sie CloudFront Origin Shield, um ursprüngliche Anfragen zu konsolidieren und zu reduzieren, wenn Sie mehrere verwenden, CDNs um die Kosteneffizienz zu erhöhen.

Weitere bewährte Methoden für die Bereitstellung von Inhalten finden Sie im [Whitepaper Inhaltsbereitstellung für Spiele](#).

GAMECOST02-BP02 Optimiere die Anzahl der Spielsitzungen, die auf jeder Spieleserverinstanz gehostet werden, um die Kosten zu optimieren

Optimieren Sie die Anzahl der pro Serverinstanz gehosteten Spielsitzungen, um eine bessere Rechenauslastung zu erreichen und die Kosten für die Recheninfrastruktur zu senken.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Um die Kosten zu optimieren, sollten Spieleentwickler die Anzahl der auf demselben physischen oder virtuellen Server gehosteten Spielsitzungen maximieren, was auch als Packungsdichte ihrer Spieleserver bezeichnet wird. Dies wird erreicht, indem die Anzahl der Spielserverprozesse erhöht wird, die gleichzeitig auf einer Instanz gehostet werden können.

Ein einzelner Spielserverprozess sollte normalerweise nicht die Nutzung der gesamten auf der EC2 Instanz verfügbaren Ressourcen erfordern. Dies ist eine der wichtigsten Methoden zur Senkung der Rechenkosten für ein Spiel und erfordert den Einsatz von Software, die mehrere Serverprozesse auf der EC2 Instance an separaten Ports starten und verwalten kann.

Amazon GameLift hat beispielsweise ein Kontingent für die maximale Anzahl von Spielserverprozessen pro Instanz festgelegt, das Sie nutzen sollten, um die Hosting-Kosten zu senken. Weitere Informationen zum aktuellen [Kontingent für maximale GameLift Spielserverprozesse pro Instance finden Sie unter Amazon Servers Endpoints and Quotas](#).

Als Alternative zur Bereitstellung von Spielserverprozessen auf virtuellen Maschinen wie EC2 Instances wird es bei Spieleentwicklern immer beliebter, ihre Spieleserver mithilfe von Container-Orchestrierungslösungen als containerbasierte Anwendungen auszuführen. Spieleentwickler können [Amazon Elastic Container Service](#) (Amazon ECS) oder [Guidance for Game Server Hosting Using Agones und Open Match auf Amazon EKS verwenden](#). Eine weitere Option ist [Game Server Hosting on AWS Fargate](#), eine serverlose Compute-Engine, die sowohl mit ECS als auch mit EKS funktioniert, sodass Sie sich auf Ihr Spiel konzentrieren können, ohne die zugrunde liegende Infrastruktur verwalten zu müssen.

Container-Lösungen bieten Funktionen zur Jobplanung, mit der automatisch eine verfügbare Container-Instance im Cluster gefunden wird, um Ihren Gameserver-Container auf der Grundlage der Ressourcenanforderungen und anderer von Ihnen festgelegter Platzierungslogik zu hosten.

Es ist jedoch wichtig, darüber nachzudenken, wie Sie die Skalierung und das Verhalten bei der Spielerplatzierung so handhaben, dass aktive Spielsitzungen nicht gestört werden.

Implementierungsschritte

- Erhöhen Sie die Packungsdichte, indem Sie mehrere Spieleserverprozesse pro EC2 Instanz mit separaten Ports und Prozessmanagement-Software ausführen.
- Verwenden Sie Amazon GameLift oder Container-Lösungen wie ECS, EKS oder, AWS Fargate um Spieleserverprozesse effizient zu verwalten und die Infrastrukturkosten zu senken.
- Überwachen Sie kontinuierlich die Ressourcennutzung, um die Packdichte zu verbessern und die Kosteneffizienz aufrechtzuerhalten, ohne das Spielerlebnis zu beeinträchtigen.

GAMECOST02-BP03 Wählen Sie die entsprechende Option zur Preisberechnung aus, um die Kosten zu senken

Führen Sie Leistungstests Ihrer Gameserver-Software für eine Vielzahl von Instanztypen und Rechenoptionen durch, um zu ermitteln, welche Option für Ihr Spiel am kostengünstigsten ist.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Neben der effizienten Nutzung der richtigen EC2 Instance-Typen für Ihren Workload sollten Sie sich auch überlegen, welche der verfügbaren Preisoptionen für Rechenleistung am besten für Ihre Kostenoptimierungsziele geeignet ist. Es stehen verschiedene Preisoptionen zur Verfügung, darunter On-Demand-Instances, Spot-Instances, Reserved Instances und Savings Plans.

[Savings Plans](#) (SPs) bieten Rabatte für Rechenleistung, indem sie Nutzungsverpflichtungen eingehen. Sie eignen sich ideal für Szenarien, in denen Sie Ihre voraussichtliche Nutzung für einen Zeitraum von 1 oder 3 Jahren nicht prognostizieren können. Sie bieten Rabatte wie Reserved Instances mit der Flexibilität, diese Rabatte für alle Regionen, Instance-Familien, Betriebssysteme und Mandanten anzuwenden. Sie können auch darauf angewendet werden AWS Fargate, wer als Gameserver-Hosting-Option für Gelegenheitsspiele oder AWS Lambda als hervorragende Option für rundenbasierte Spiele genutzt wird, für die keine Spieleserver erforderlich sind. Weitere Informationen findest du unter [Entwicklung eines serverlosen Mehrspieler-Spiels](#), das skalierbar ist.

Savings Plans werden bei der Veröffentlichung von Spielen eingeführt, um Kosten für die Workloads der Spieleserver zu sparen, die zu den Ausgaben der EC2 Instanzen beitragen, wenn das Spiel für das Publikum veröffentlicht wird. Savings Plans können auch nach der Veröffentlichung eingeführt

werden, wenn das Game Operations Team einen besseren Überblick über den Spielerverkehr hat, nachdem das Spiel über einen längeren Zeitraum in Produktion war.

Da Savings Plans regionale Flexibilität bieten, sind sie besonders ideal, um die Ausgaben von Spieleservern für Spiele mit unvorhersehbarer Nutzung in verschiedenen Regionen zu optimieren.

Wenn Ihre tägliche Spielernutzung beispielsweise mindestens 20 Server zur Unterstützung Ihrer Spielerbasis erfordert, in regelmäßigen Abständen jedoch bis zu 40 Server benötigt, sollten Sie erwägen, Savings Plan abzuschließen, um den Grundwert der 20 Server abzudecken, da dieser Nutzungsbedarf vorhersehbar und konsistent ist und zu einer maximalen Nutzung der von Ihnen gekauften Nutzungsvereinbarung führt.

Maximieren Sie die Nutzung von Savings Plans und erweitern Sie sie um weitere Kaufoptionen, die mehr Flexibilität für unvorhersehbare Auslastungsspitzen bei Spieleservern bieten, wie z. B. On-Demand-Instances und Spot-Instances, um optimale Einsparungen zu erzielen.

Spot-Instances eignen sich ideal für den Betrieb von Spieleservern, da sie die größten Rechenrabatte bieten, keine Nutzungsverpflichtungen erfordern und Flexibilität für unvorhersehbare und stark beanspruchte Workloads bieten. Spot-Instances können jedoch unterbrochen werden und eignen sich daher am besten für Spielservers-Workloads mit kurzer Spielesitzungsdauer oder für Situationen, in denen die Toleranz gegenüber Unterbrechungen höher ist.

Weitere Informationen zu Anleitungen für den Betrieb von Spieleservern mit Kubernetes auf Amazon EKS mit EC2 Spot-Instances finden Sie unter [So führen Sie Massively Multiplayer-Spiele mit EC2 Spot mithilfe von Aurora Serverless](#) aus.

Verwenden Sie [Amazon EC2 Spot-Instances](#), um Pools mit der geringsten Wahrscheinlichkeit einer Unterbrechung zu ermitteln, die im Vergleich zu On-Demand-Tarifen maximale Einsparungen ermöglichen.

Wenn Sie Spot verwenden, wird außerdem empfohlen, Spielservers-Workloads über mehrere EC2 Instance-Typen und Availability Zones hinweg auszuführen, um Ihre Kapazitätsnutzung AWS-Region zu diversifizieren und das Risiko von Unterbrechungen zu reduzieren.

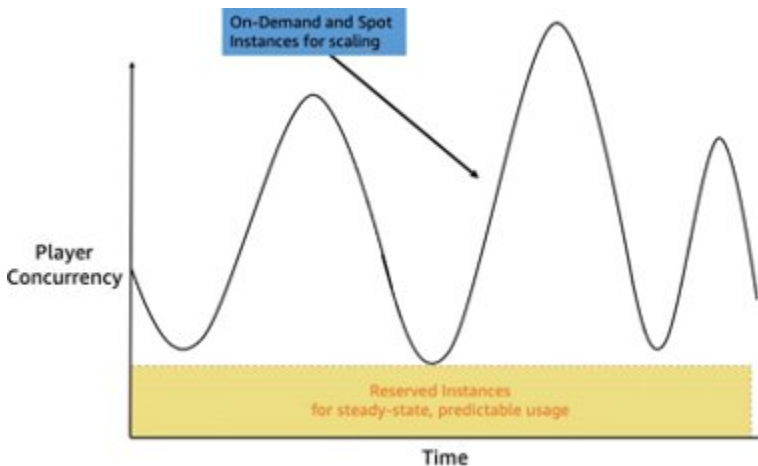
Erwägen Sie die Verwendung von Spot-Instances in Kombination mit On-Demand-Instances, um die Auswirkungen potenzieller Unterbrechungen aktiver Spielsitzungen zu minimieren, und nutzen Sie eine kapazitätsoptimierte Zuweisungsstrategie, um das Risiko von Unterbrechungen weiter zu verringern.

Weitere bewährte Methoden finden Sie in den [Best Practices für Amazon EC2 Spot](#). Der [Kapazitätsausgleich in Auto Scaling als Ersatz für gefährdete Spot-Instances](#) kann verwendet

werden, um proaktiv zu überwachen und zusätzliche Kapazität hinzuzufügen, wenn Spot-Instances einem erhöhten Ausfallrisiko ausgesetzt sind.

[Amazon GameLift FleetIQ lässt](#) sich in Spot-Instances integrieren, um die Nutzung kostengünstiger Spot-Instances zu optimieren und gleichzeitig das Risiko von Unterbrechungen zu reduzieren. Wenn du dein Spiel über hostest GameLift, sieh dir die GameLift Dokumentation zur Auswahl der Computerressourcen an. Weitere Informationen findest du unter [Computerressourcen für eine verwaltete Flotte auswählen](#).

Das folgende Diagramm veranschaulicht anhand eines Beispiels die Verwendung mehrerer Preisoptionen für Rechenleistung für Spieleserver-Workloads:



Hosten von Spieleservern mit mehreren EC2 Preisoptionen

In der Abbildung schwankt die Parallelität der Spieler im Laufe der Zeit, was es schwierig macht, die Auslastung zu verwalten und die Kosten zu optimieren. Um dieser Fluktuation entgegenzuwirken, sollten Sie erwägen, eine Mischung aus verschiedenen Preisoptionen für Rechenleistung zu wählen. Verwenden Sie Savings Plans, EC2 um die Anforderungen Ihrer Mindestnutzungsanforderungen zu erfüllen, und setzen Sie auf EC2 On-Demand-Instances und EC2 Spot-Instances, um den Anforderungen Ihrer Spieler gerecht zu werden.

Implementierungsschritte

- Verwenden Sie Savings Plans für eine vorhersehbare Basisnutzung und kombinieren Sie sie mit Spot- und On-Demand-Instances für Flexibilität und Kostenoptimierung bei Nutzungsspitzen.
- Verwenden Sie Spot-Instances für Spieleserver mit kurzer Sitzungsdauer oder höherer Unterbrechungstoleranz und diversifizieren Sie die Instance-Typen und Availability Zones, um das Risiko zu minimieren.

- Implementieren Sie Tools wie EC2 Spot Instances Advisor, Capacity Rebalancing und GameLift FleetIQ, um die Nutzung von Spot-Instances zu optimieren und Unterbrechungen proaktiv zu managen.

Datenübertragungskosten

GAMECOST03: Wie optimieren Sie die Datenübertragungskosten für Ihre Spieleinfrastruktur?

Spiele können eine beträchtliche Menge an Daten über das Internet zwischen den Spielclient-Geräten deiner Spieler und deiner Spieleinfrastruktur übertragen, um das Spielerlebnis zu gewährleisten, sowie zwischen den Komponenten deiner Spieleinfrastruktur.

Datenübertragungen finden beispielsweise statt, wenn Spieler Updates für Spielinhalte auf ihre Spielclients herunterladen, ihren Spielfortschritt in der Cloud speichern, mit ihren Freunden an Echtzeit-Multiplayer-Spielsitzungen teilnehmen und wenn Ihre Spielinfrastruktur Daten zwischen Regionen und Availability Zones überträgt. Es ist wichtig zu wissen, wo die Datenübertragung innerhalb deiner Spiellast stattfindet, um deine Architekturauswahl zu optimieren und diese Datenübertragungskosten zu reduzieren.

Beachten Sie die folgenden bewährten Methoden, um die Datenübertragungskosten für Ihre Spiellast zu optimieren:

Best Practices

- [GAMECOST03-BP01 Wählen Sie den geeigneten Speichertyp für benutzergenerierte Inhalte, um die Kosten zu senken](#)
- [GAMECOST03-BP02 Optimiere Datenbanken für Spiele-Backends](#)

GAMECOST03-BP01 Wählen Sie den geeigneten Speichertyp für benutzergenerierte Inhalte, um die Kosten zu senken

Jeder Datentyp, der in Ihrem Spiel generiert und gespeichert wird, hat einzigartige Eigenschaften, die Sie bei der Auswahl der richtigen Speicherlösung für Ihre Arbeitslast berücksichtigen sollten.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Niedrig

Implementierungsleitfaden

Verwenden Sie Amazon S3 Object Lifecycle Management, um Objektdaten in der kostengünstigsten Speicherklasse zu speichern. Amazon S3 bietet mehrere [Speicherklassen](#) und [Object Lifecycle Management](#), um die Einrichtung einfacher und detaillierter Richtlinien für die automatische Übertragung von Daten zwischen Speicherstufen zur Kostensenkung zu vereinfachen. Anstatt Daten standardmäßig einfach in der S3-Standard Speicherklasse zu speichern, sollten Sie eine Lebenszykluskonfiguration einrichten, um Daten im Laufe der Zeit automatisch zwischen den Stufen zu übertragen, oder die Speicherklasse S3 Intelligent-Tiering für unbekannte oder sich ändernde Zugriffsmuster verwenden.

Alternativ kann S3 Intelligent-Tiering Daten kostengünstig und automatisch zwischen Stufen übertragen. Es wird als Standard Speicherklasse empfohlen, da es eine Kostenoptimierung ermöglicht, ohne dass Lebenszyklusrichtlinien manuell eingerichtet werden müssen, und ist jetzt die beste Wahl für kleine und kurzlebige Objekte. Weitere Informationen finden Sie unter [Amazon S3 Intelligent-Tiering — Verbesserte Kostenoptimierungen für kurzlebige](#) und kleine Objekte.

Zu den häufigsten Anwendungsfällen für Amazon S3 gehören die Speicherung von Spielressourcen, statischen Inhalten, Spielprotokollen, Data-Lake-Speicher und Backups. Für Anwendungsfälle, in denen Dateisysteme erforderlich sind, wie z. B. das Anhängen gemeinsam genutzter Dateisysteme an Arbeitsstationen während der Entwicklung, sollten Sie [Amazon Elastic File System \(Amazon EFS\)](#) in Betracht ziehen, das verschiedene Speicherklassen bietet und automatisch wächst und schrumpft, wenn Sie Dateien hinzufügen und entfernen, ohne dass die Infrastruktur verwaltet werden muss.

[Amazon S3 One Zone](#) -IA ist eine ideale Speicheroption für vorübergehende Daten im Zusammenhang mit Spielsitzungen, Matchmaking oder anderen kurzlebigen Informationen, die bei Bedarf neu erstellt werden können. Für diese Art von Spieldaten ist keine Redundanz über mehrere Availability Zones hinweg erforderlich (). AZs Diese kostengünstigere Speicherklasse eignet sich gut für Aufzeichnungen von Spieleraktionen, Spielereignissen und anderen Telemetriedaten, die für Analysen oder Debugging verwendet werden.

Der wichtigste Vorteil bei der Kostenoptimierung durch die Verwendung von S3 Express One Zone für solche Spieldaten sind die erheblichen Kosteneinsparungen im Vergleich zur Standard Speicherklasse S3 mit einer Reduzierung der Speicherkosten um bis zu 20%. Dies kann besonders vorteilhaft für Spiele mit großen Datenmengen sein, die nicht das gleiche Maß an Haltbarkeit und Verfügbarkeit erfordern wie unternehmenskritische Anwendungsdaten. Durch die Nutzung von S3 One Zone können Spieleentwickler und Herausgeber ihre Cloud-Speicherkosten optimieren, ohne das allgemeine Spielerlebnis zu beeinträchtigen.

Implementierungsschritte

- Konfigurieren Sie Amazon S3 S3-Lebenszyklusrichtlinien, um Daten zwischen Speicherklassen zu übertragen, oder verwenden Sie S3 Intelligent-Tiering als Standard für die automatische Kostenoptimierung bei wechselnden Zugriffsmustern.
- Verwenden Sie S3 One Zone-Infrequent Access für vorübergehende Spielsitzungsdaten wie Telemetrie- und Matchmaking-Aufzeichnungen, um die Speicherkosten um bis zu 20% zu senken und gleichzeitig eine ausreichende Verfügbarkeit aufrechtzuerhalten.
- Wenn Sie während der Entwicklung gemeinsam genutzte Dateisysteme benötigen, verwenden Sie Amazon EFS, um das Speichermanagement mit elastischer Kapazität und mehreren Speicherklassen zu vereinfachen.

GAMECOST03-BP02 Optimiere Datenbanken für Spiele-Backends

Spiele sind stark auf Datenbanken angewiesen, um eine Vielzahl kritischer Daten zu speichern, von Spielerprofilen und Inventaren bis hin zu Mikrotransaktionen und Fortschrittsstatistiken im Spiel. Datenbanken spielen auch eine entscheidende Rolle bei der Verwaltung der sozialen Aspekte von Spielen, z. B. bei der Bildung und Pflege von Spielergruppen und Partys und der Durchsetzung von Moderationsrichtlinien. Wenn die Spielerbasis eines Spiels wächst, werden die damit verbundenen Datenbankkosten unweigerlich steigen, um den steigenden Daten- und Nutzungsanforderungen gerecht zu werden.

Risikostufe bei fehlender Befolgung dieser bewährten Methode: Mittel

Implementierungsleitfaden

Für Spiele-Backends, die auf Amazon Aurora laufen, gibt es mehrere Strategien zur Kostenoptimierung, die eingesetzt werden können. Eine wichtige Empfehlung besteht darin, [Ihre Read Replicas auf der Grundlage von Nutzungsmustern auto](#) zu skalieren und dabei die Anzahl der Replikate dynamisch nach oben oder unten zu skalieren, um Verkehrsschwankungen zu bewältigen. Das bedeutet, dass Sie für die Ressourcen zahlen, die Sie wirklich benötigen. Eine weitere Optimierungstaktik besteht darin, Read Replicas, die für Spieleanalysen verwendet werden, durch Exporte von DB-Snapshots nach Amazon S3 zu ersetzen, da der S3-Speicherservice im Allgemeinen günstiger ist als bereitgestellte Aurora-Datenbank-Instances. Weitere Informationen finden Sie unter [Exportieren von DB-Snapshot-Daten nach Amazon S3 für Amazon RDS](#).

Die Nutzung [reservierter DB-Instances für Amazon Aurora](#) für Ihre Kerndatenbank-Instances und die Umstellung auf die [Aurora-Serverless-Konfiguration](#) können auch zu erheblichen

langfristigen Kosteneinsparungen führen, da Sie mehr Flexibilität und [detaillierte Kontrolle über Ihre Ressourcennutzung](#) bieten.

In ähnlicher Weise kann für Spiele-Backends, die Amazon DynamoDB verwenden, der [DynamoDB-On-Demand-Kapazitätsmodus](#) eine effektive Wahl sein, insbesondere bei neuen oder unvorhersehbaren Workloads, da Sie so nur für die Ressourcen zahlen müssen, die Sie verbrauchen, ohne dass Sie zu viel bereitstellen müssen. Wenn Ihre Datenverkehrsmuster im Spiel im Laufe der Zeit stabiler und vorhersehbarer werden, können Sie dann in den [Modus für bereitgestellte Kapazität von DynamoDB wechseln, der durch eine bessere Kapazitätsplanung](#) zu Kosteneinsparungen führen kann. Die Aktivierung der auto-scaling für Ihre DynamoDB-Tabellen ist eine weitere wichtige Optimierung, die es dem Service ermöglicht, die bereitgestellte Kapazität dynamisch an Verkehrsschwankungen anzupassen. Testen Sie die Datenstruktur Ihres Spiels vor dem Start in einer Entwicklungsumgebung, um unnötige [lokale Sekundärindizes \(LSIs\)](#) und [globale Sekundärindizes \(\)](#) zu finden und zu entfernen. GSIs Dies kann zu erheblichen Kosteneinsparungen bei der Speicherung und dem Betrieb von Spieldaten führen. Wenn Sie [ineffiziente Scanvorgänge](#) aus Ihrem Spiel-Backend-Code zugunsten gezielterer Abfragen entfernen, [reservierte Amazon DynamoDB-Kapazität erwerben und DynamoDB Streams mit AWS Lambda Triggern zur Verarbeitung von Spiel-Backend-Ereignissen nutzen, können Sie Ihre DynamoDB-Kosten](#) weiter optimieren. Weitere Informationen finden Sie unter [Bewährte Methoden für das Abfragen und Scannen von Daten in DynamoDB](#).

Durch die Implementierung dieser Strategien zur Kostenoptimierung sowohl für Amazon Aurora als auch für DynamoDB können Spieleentwickler und Herausgeber ihre Ausgaben für Spiele-Backend-Datenbanken erheblich reduzieren.

Implementierungsschritte

- Verwenden Sie die auto-scaling von Aurora Read Replica und DB-Snapshot-Exporte nach Amazon S3, um schwankenden Traffic- und Analyseanforderungen kosteneffizient zu bewältigen.
- Optimieren Sie die DynamoDB-Kosten, indem Sie mit On-Demand-Kapazität für neue Workloads beginnen, auf bereitgestellte Kapazität mit auto-scaling für vorhersehbaren Traffic umsteigen und ungenutzte und entfernen. LSIs GSIs
- Vermeiden Sie ineffiziente Scanvorgänge zugunsten gezielter Abfragen, verwenden Sie Reserved Instances oder Reserved Capacity und verwenden Sie DynamoDB Streams mit AWS Lambda für die Ereignisverarbeitung.

Verwaltung von Nachfrage- und Angebotsressourcen

Für die Verwaltung von Nachfrage- und Angebotsressourcen gibt es keine spezifischen Best Practices für Games Lens.

Weitere Informationen zur Steuerung der Nachfrage und zur Bereitstellung von Ressourcen finden Sie unter [Cost Optimization Pillar — AWS Well-Architected Framework](#).

Optimierung im Laufe der Zeit

Es gibt keine spezifischen Best Practices für die Optimierung im Laufe der Zeit für Games Lens.

Weitere Informationen zur Optimierung der Kosten im Laufe der Zeit finden Sie unter [Cost Optimization Pillar — AWS Well-Architected Framework](#).

Ressourcen

Weitere Informationen zu bewährten Methoden zur Kostenoptimierung finden Sie in den folgenden Ressourcen:

Zugehörige Dokumente:

- [Wie senke ich die Datenübertragungsgebühren für mein NAT-Gateway in Amazon VPC?](#)
- [Vorstellung des Amazon GameLift FleetIQ-Adapters für Agones](#)
- [Wie finde ich die wichtigsten Beiträge zum NAT-Gateway-Traffic in meiner Amazon VPC?](#)
- [Wählen Sie die richtige Rechenstrategie für Ihre globalen Spieleserver](#)
- [AWS Well-Architected Labs — Kostengünstige Ressourcen](#)
- [Das Amazon VPC CNI-Plugin erhöht die Limits für Pods pro Knoten](#)
- [Bewährte Architekturpraktiken zur Kostenoptimierung](#)
- [Reduzierung der Wartezeiten der Spieler und richtige Dimensionierung der Rechenzuweisung mithilfe von Amazon SageMaker AI RL und Amazon EKS](#)
- [AWS Compute Optimizer](#)
- [Electronic Arts optimiert Speicherkosten und Betriebsabläufe mithilfe von Amazon S3 Intelligent-Tiering und Amazon Glacier](#)
- [Vermeiden Sie unfreundliche Lizenzierungspraktiken, indem Sie Windows-Workloads auf Linux migrieren](#)

- [Überblick über die Datenübertragungskosten für gängige Architekturen](#)
- [AWS und Kubecost arbeiten zusammen, um EKS-Kunden die Kostenüberwachung zu ermöglichen](#)
- [Den Grundstein legen: Richten Sie Ihre Umgebung für die Kostenoptimierung ein](#)
- [Überblick über Amazon EC2 Spot-Instances](#)

Nachhaltigkeit

Die Säule Nachhaltigkeit bietet Gestaltungsprinzipien, betriebliche Leitlinien, bewährte Verfahren und Verbesserungspläne, mit denen Sie die Nachhaltigkeitsziele für Ihre AWS Workloads erreichen können.

Implementierungsleitlinien zur Implementierung finden Sie im Whitepaper [Sustainability Pillar — AWS Well-Architected Framework](#).

Schwerpunktbereiche

- [Designprinzipien](#)
- [Auswahl der Region](#)
- [Ausrichtung am Bedarf](#)
- [Software und Architektur](#)
- [Datenverwaltung](#)
- [Hardware und Services](#)
- [Ressourcen](#)

Designprinzipien

Die Nachhaltigkeit der Spielebranche entwickelt sich in verschiedenen Regionen der Welt unterschiedlich schnell. Angesichts der nachhaltigen Energieversorgung in weiten Teilen Nordamerika und der Nachhaltigkeitsvorschriften in der EU und im Vereinigten Königreich verfolgen Architekten mehrere Ansätze, um in naher future umweltfreundlichere Arbeitslasten zu erreichen. Die [Nachhaltigkeitssäule](#) Well-Architected kann genutzt werden, um diese Maßnahmen für eine Vielzahl von Workloads zu erreichen.

In diesem Abschnitt der Linse werden mehrere bewährte Methoden beschrieben, die für Workloads in Spielen verwendet werden können.

- Wählen Sie den geeigneten Speichertyp für Benutzerdaten von Spielen aus.
- Beachten Sie die Richtlinien für den Datenlebenszyklus, mit denen Daten dedupliziert und nicht benötigte Daten aus Ihren Workloads entfernt werden.
- Seien Sie bei der richtigen Dimensionierung der Bereitstellung von Rechenressourcen wählerisch.

- Verwenden Sie Serverless für kurze und transaktionale Prozesse.

Auswahl der Region

Es gibt keine spezifischen Best Practices für die [Regionsauswahl](#) für Games Lens. Weitere Informationen finden Sie unter [Säule Nachhaltigkeit — AWS Well-Architected Framework](#).

Ausrichtung am Bedarf

Es gibt keine [Ausrichtung auf die Forderung nach](#) bewährten Verfahren, die speziell für Games Lens gelten. Weitere Informationen finden Sie unter [Säule Nachhaltigkeit — AWS Well-Architected Framework](#).

Software und Architektur

Es gibt keine spezifischen Best Practices für [Software und Architektur](#) für Games Lens. Weitere Informationen finden Sie unter [Säule Nachhaltigkeit — AWS Well-Architected Framework](#).

Datenverwaltung

GAMESUS01: Wie verwaltest du Benutzer- und Spieldaten in deinem Spielsystem?

Entwickeln Sie eine Strategie für den Datenlebenszyklus, die Datenspeicherung und Relevanz optimiert, indem nur kritische historische Daten aufbewahrt werden.

Best Practices

- [GAMESUS01-BP01 Verwenden Sie Speichertechnologien, die den Mustern entsprechen, die an Benutzerinhalte, Abonnenteninformationen und Käufe im Spiel angepasst sind](#)
- [GAMESUS01-BP02 Verwenden Sie Lebenszyklusrichtlinien oder TTL-Ablaufzeiten, um nicht benötigte Spielbenutzerdaten, Protokolldateien oder veraltete Inhalte zu löschen](#)

GAMESUS01-BP01 Verwenden Sie Speichertechnologien, die den Mustern entsprechen, die an Benutzerinhalte, Abonnenteninformationen und Käufe im Spiel angepasst sind

Sie sollten Ihre Daten nach Art, Aufbewahrungsbedarf und Häufigkeit des Zugriffs klassifizieren. Auf diese Weise können Sie die am besten optimierte Speicherlösung für die unzähligen Datentypen auswählen, die Ihr Spiel oder Ihre Backend-Dienste produzieren. Daten, die sich schnell ändern, sollten in Key-Value- oder In-Memory-Datenbankdiensten gespeichert werden. Transaktionsdaten sollten in relationalen Datenbankdiensten gespeichert werden. Große Dateien, Spielressourcen oder benutzergenerierte Inhalte sollten in Objektspeicherdiensten gespeichert werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Spiele erzeugen und verbrauchen eine Vielzahl von Datentypen, für die Speicherlösungen erforderlich sind, die für Zugriffshäufigkeit, Latenz und Kosten optimiert sind. Gespeicherte Daten sollten mithilfe von Tags klassifiziert werden, um zwischen Daten zu unterscheiden, die entfernt werden können oder langfristig gespeichert werden müssen.

Die folgenden Dienste eignen sich gut für eine Vielzahl von Anwendungsfällen für Spiele:

[Amazon Aurora](#) (kompatibel mit MySQL und PostgreSQL) bietet hohe Verfügbarkeit, geringe Latenz und automatische Skalierung und ist damit eine hervorragende Wahl für den Umgang mit großen Mengen an Transaktionsdaten, wie z. B. Verwaltung und Authentifizierung von Spielerkonten, In-Game-Ökonomien, Bestenlisten und Spieler-Rankings, Persistenz des Spielstatus, Event- und Kampagnenmanagement sowie Bereitstellung in mehreren Regionen und mit hoher Verfügbarkeit.

[Amazon DynamoDB](#) ist eine vollständig verwaltete NoSQL-Datenbank, die für ihre geringe Latenz, ihren hohen Durchsatz und ihre nahtlose Skalierbarkeit bekannt ist. Dadurch eignet sie sich ideal für die Verarbeitung von Spielerdaten in Echtzeit, Sitzungsmanagement, Inventar, In-Game-Wirtschaft, Echtzeit-Multiplayer-Spielstatus, Matchmaking, Ereignisprotokollierung und Skalierung für ein globales Publikum.

[Amazon DocumentDB](#) (kompatibel mit MongoDB) bietet einen skalierbaren dokumentenorientierten Datenbankservice mit niedriger Latenz, der sich perfekt für die Speicherung flexibler, halbstrukturierter Daten wie Inventarsystem, Spielerprofile und Anpassungen, Spielwelten und prozedural generierte Inhalte, soziale Interaktionen und Spielerinteraktionen, Analysen und Verhaltensverfolgung sowie Metadaten und Konfigurationen im Spiel eignet.

[Amazon ElastiCache](#) unterstützt In-Memory-Caching mit Redis oder Memcached und bietet so schnellen Datenzugriff und kürzere Reaktionszeiten. Dies ist entscheidend für Echtzeit-Multiplayer-Spiele, bei denen Geschwindigkeit und Leistung für ein reibungsloses Benutzererlebnis unerlässlich sind. ElastiCache wird in Spielen für Echtzeit-Bestenlisten, Sitzungsmanagement, Zwischenspeicherung von Spielmetadaten, Chat und Messaging im Spiel, Spielersuche, Echtzeitanalysen und Telemetrie sowie Skalierung für Ereignisse mit hohem Besucheraufkommen verwendet.

[Amazon Simple Storage Service \(S3\)](#) kann verwendet werden, um Objekte wie Spielinhalte, Videos, Bilder, Textprotokolldateien und mehr zu speichern. S3 ist ein Objektspeicherservice, der branchenführende Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung bietet.

Wenn er mehrere Speicherklassen bietet, die häufigen und seltenen Datenzugriff unterstützen, sowie kostengünstigen Archivspeicher. Für Daten, auf die während der Entwicklung häufig zugegriffen wird, sollten Studios Objekte im [S3-Standard](#) speichern, um eine geringe Latenz und eine hohe Durchsatzleistung zu gewährleisten. Bei Daten, die häufig von heiß nach kalt oder umgekehrt wechseln, sollten Studios [S3 Intelligent-Tiering](#) in Betracht ziehen. Intelligent-Tiering überwacht die Zugriffsmuster Ihrer Daten und verschiebt Daten automatisch auf die kostengünstigste Zugriffsebene.

Studios, die einen hohen Durchsatz und eine geringe Latenz benötigen und damit einverstanden sind, in einer einzigen Availability Zone zu leben, sollten [S3 Express One Zone](#) verwenden. Dadurch werden Daten auf eine einzige AZ repliziert und die Datenzugriffsgeschwindigkeit kann im Vergleich zum S3-Standard verbessert werden. Für umfangreiche Archivierungsanforderungen historischer Daten bietet Amazon auch [Amazon Glacier an](#). Die Amazon Glacier-Speicherklassen wurden speziell für die Datenarchivierung entwickelt und bieten Ihnen hohe Leistung, Flexibilität beim Abruf und kostengünstigen Archivspeicher in der Cloud.

[Amazon Elastic Block Store](#) kann verwendet werden, um die Binärdateien, ausführbaren Dateien und Konfigurationen von Spieleservern zu speichern, die Ihre Spieleserver oder Asset-Repositorys benötigen, um zu funktionieren. Sie sollten ungenutzte Volumes, die nicht an eine Instance angehängt sind, als Snapshot speichern und löschen. EC2 Dadurch sparen Sie sich die anfallenden Speichergebühren und reduzieren gleichzeitig die Nutzung nicht benötigter Dienste und Hardware.

Implementierungsschritte

- Klassifizieren Sie Spieldaten nach Typ, Aufbewahrungsbedarf und Zugriffshäufigkeit und kennzeichnen Sie die Daten, um zwischen kurzfristigen und langfristigen Speicheranforderungen zu unterscheiden.

- Verwenden Sie Amazon Aurora für Transaktionsdaten, DynamoDB für Spielerdaten in Echtzeit, DocumentDB für halbstrukturierte Daten und ElastiCache für das Zwischenspeichern zeitkritischer Spielinformationen mit niedriger Latenz.
- Speichern Sie Spielressourcen, Protokolle und benutzergenerierte Inhalte in Amazon S3, wählen Sie geeignete Speicherklassen (z. B. Intelligent-Tiering, One Zone und Glacier) auf der Grundlage von Zugriffsmustern und Archivierungsanforderungen aus und verwenden Sie EBS für Gameserver-Binärdateien und -konfigurationen mit regulärer Snapshot-Verwaltung.

GAMESUS01-BP02 Verwenden Sie Lebenszyklusrichtlinien oder TTL-Ablaufzeiten, um nicht benötigte Spielbenutzerdaten, Protokolldateien oder veraltete Inhalte zu löschen

Sie können Tags und Datentypen verwenden, um Lebenszyklusrichtlinien oder TTLs zu erstellen, um Daten in den Archivspeicher zu verschieben oder sie vollständig aus dem Dienst zu entfernen. Dazu können temporäre Konfigurationen, abgelaufene archivierte Inhalte und historische Protokolle gehören, die nicht mehr benötigt werden. Die meisten Dienste unterstützen Tagging.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Für in S3 gespeicherte Daten können Sie Lebenszyklusrichtlinien verwenden, um die Daten auf Speicherebenen mit seltenem Zugriff und Archivierung zu verschieben. Sie können in einer S3-Lebenszykluskonfiguration Regeln definieren, um Objekte aus einer Speicherklasse zu einer anderen Speicherklasse zu überführen, um Speicherkosten zu sparen. Wenn Sie die Zugriffsmuster Ihrer Objekte nicht kennen oder sich Ihre Zugriffsmuster über die Zeit verändern, können Sie die Objekte zur Speicherklasse S3 Intelligent-Tiering überführen, um automatische Kosteneinsparungen zu erzielen.

Amazon S3 unterstützt das Wasserfallmodell für die Überführung zwischen Speicherklassen wie im folgenden Diagramm gezeigt.

Sie können zu Ihrer S3-Lebenszykluskonfiguration Übertragungsaktionen hinzufügen und Amazon S3 damit anweisen, Objekte am Ende ihrer Lebensdauer zu löschen. Wenn ein Objekt aufgrund seiner Lebenszykluskonfiguration das Ende seiner Lebensdauer erreicht, führt Amazon S3 eine Ablaufaktion aus, die darauf basiert, in welchem S3-Versionierungsstatus sich der Bucket befindet:

- Bucket ohne Versionierung: Amazon S3 stellt das Objekt zur Entfernung in die Warteschlange und entfernt es asynchron, wodurch das Objekt dauerhaft entfernt wird.
- Bucket mit aktivierter Versionierung: Wenn es sich bei der aktuellen Objektversion nicht um eine Löschmarkierung handelt, fügt Amazon S3 eine Löschmarkierung mit einer eindeutigen Versions-ID hinzu. Damit ist die aktuelle Version nicht mehr aktuell und die Löschmarkierung wird zur aktuellen Version.
- Bucket mit angehaltener Versionierung: Amazon S3 erstellt eine Löschmarkierung mit Null als Versions-ID. Diese Löschmarkierung ersetzt eine Objektversion durch eine Null-Versions-ID in der Versionshierarchie, wodurch das Objekt effektiv gelöscht wird.
- Wenn Sie einem Bucket eine Lebenszykluskonfiguration hinzufügen, gelten die Konfigurationsregeln für vorhandene Objekte und für Objekte, die Sie später hinzufügen. Wenn Sie beispielsweise heute eine Lebenszyklus-Konfigurationsregel mit einer Ablaufaktion hinzufügen, die bewirkt, dass Objekte mit einem bestimmten Präfix 30 Tage nach ihrer Erstellung ablaufen, stellt Amazon S3 eine Warteschlange für die Entfernung vorhandener Objekte, die älter als 30 Tage sind und das angegebene Präfix haben, in die Warteschlange.

Time To Live (TTL) für DynamoDB ist eine kosteneffiziente Methode zum Löschen von Elementen, die nicht mehr relevant sind. TTL ermöglicht es Ihnen, einen Ablaufzeitstempel pro Element zu definieren, um festzulegen, wann ein Element nicht mehr benötigt wird. DynamoDB löscht abgelaufene Elemente automatisch innerhalb weniger Tage nach ihrem Ablauftermin, ohne dass dafür Schreibdurchsatz verbraucht wird.

- Um TTL zu verwenden, aktivieren Sie es zunächst in einer Tabelle. Definieren Sie dann ein bestimmtes Attribut zum Speichern des TTL-Ablaufzeitstempels. Der Zeitstempel muss im [Unix-Zeitformat](#) in Sekunden gespeichert werden. Immer, wenn ein Element erstellt oder aktualisiert wird, können Sie die Ablaufzeit berechnen und sie im TTL-Attribut speichern.
- Elemente mit gültigen, abgelaufenen TTL-Attributen können vom System gelöscht werden, normalerweise innerhalb weniger Tage nach ihrem Ablauf. Sie können die abgelaufenen Elemente, deren Löschung noch aussteht, weiterhin aktualisieren. Dazu gehören auch das Ändern oder Entfernen ihrer TTL-Attribute. Wenn Sie ein abgelaufenes Element aktualisieren, empfiehlt es sich, einen Bedingungsausdruck zu verwenden, um sicherzustellen, dass das Element anschließend nicht gelöscht wurde. Verwenden Sie Filterausdrücke, um abgelaufene Elemente aus [Scan-](#) und [Abfrageergebnissen](#) zu entfernen.
- Gelöschte Elemente funktionieren ähnlich wie Objekte, die bei typischen Löschvorgängen gelöscht wurden. Nach dem Löschen gehen Elemente als Dienstlöschungen statt als Benutzerlöschungen in

DynamoDB Streams und werden genau wie andere Löschvorgänge aus lokalen Sekundärindizes und globalen Sekundärindizes entfernt.

Mit ElastiCache for Redis können Sie die Aktualität Ihrer zwischengespeicherten Daten kontrollieren, indem Sie zwischengespeicherte Schlüssel oder deren Ablauf verwenden. Nach Ablauf der festgelegten Zeit wird der Schlüssel aus dem Cache gelöscht, und der Zugriff auf den Quelldatenspeicher ist ebenso erforderlich wie der Zugriff auf die aktualisierten Daten.

- Zwei Prinzipien bestimmen, welche Caching-Muster geeignet TTLs sind und welche Typen von zu implementieren sind. Zunächst ist es wichtig, dass Sie die Änderungsrate der zugrunde liegenden Daten verstehen. Zweitens ist es wichtig, dass Sie das Risiko abwägen, dass veraltete Daten an Ihre Anwendung zurückgegeben werden und nicht die aktualisierten Daten.
- Bei dynamischen Daten, die sich häufig ändern, sollten Sie möglicherweise niedrigere Werte anwenden TTLs, sodass die Daten mit einer Änderungsrate ablaufen, die der der Primärdatenbank entspricht. Dadurch wird das Risiko verringert, dass veraltete Daten zurückgegeben werden, während gleichzeitig ein Puffer zum Auslagern von Datenbankanforderungen bereitgestellt wird.
- Es ist auch wichtig zu wissen, dass, auch wenn Sie Daten nur für Minuten oder Sekunden zwischenspeichern und nicht für längere Zeiträume, eine angemessene Anwendung Ihrer zwischengespeicherten Schlüssel TTLs zu einer Leistungssteigerung und einem insgesamt besseren Spielerlebnis mit Ihrem Spiel führen kann.

Implementierungsschritte

- Verwenden Sie Amazon S3 Lifecycle-Richtlinien, um Objekte auf seltene Zugriffs- oder Archivierungsstufen umzustellen und Ablaufaktionen zu konfigurieren, um unnötige Objekte auf der Grundlage von Lebenszyklusregeln zu löschen.
- Aktivieren Sie Time to Live (TTL) in DynamoDB-Tabellen, um abgelaufene Elemente automatisch zu löschen, ohne den Schreibdurchsatz zu beanspruchen, und definieren Sie den Ablaufzeitstempel in der Unix-Epochezeit.
- Je nach Datenänderungsrate und Risikotoleranz TTLs für veraltete Daten werden die ElastiCache Schlüssel entsprechend eingestellt, sodass die Aktualität der zwischengespeicherten Daten gewährleistet und das Spielerlebnis verbessert wird.

Hardware und Services

GAMESUS02: Wie verwalten Sie die Nutzung von Rechenressourcen im Backend Ihrer Spiele?

Studios sollten eine Rechenstrategie entwickeln, die eine Mischung aus verschiedenen Rechenarten, verwalteten Diensten und Sparplänen verwendet, um Ihre Nutzung zu optimieren. Sie sollten auch optimieren, wie Spieleserver und Backend-Dienste auf Recheninstanzen gepackt werden, um die Anzahl nicht benötigter Ressourcen zu reduzieren.

Best Practices

- [GAMESUS02-BP01 Wählen Sie Managed Services für geeignete Rechen-Workloads](#)
- [GAMESUS02-BP02 Passen Sie die richtige Größe Ihrer Rechenleistung an und setzen Sie GPU-Leistung nur dort ein, wo sie benötigt wird](#)

GAMESUS02-BP01 Wählen Sie Managed Services für geeignete Rechen-Workloads

Richten Sie Ihre Spiele-Back-End-Dienste so ein, dass sie Managed Services für ereignisgesteuerte oder stark variable Traffic-Workloads verwenden. Managed Services verlagern die Verwaltung der Infrastruktur auf mehrere Nutzer AWS und verteilen die Umweltbelastung auf mehrere Nutzer, da die Steuerungsebenen von mehreren Mandanten genutzt werden.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

AWS Dienste wie AWS Lambda AWS Fargate (Containers) und Amazon Gamelift (Gameserver-Orchestrierung) können Code und Container ausführen oder Ihre Spieleserver orchestrieren, ohne die zugrunde liegende Infrastruktur verwalten zu müssen. Diese Dienste werden automatisch auf der Grundlage der Nachfrage der Spieler skaliert, und Ihnen werden nur die Ressourcen in Rechnung gestellt, die Sie verbrauchen. Da die zugrunde liegende Infrastruktur in Ihrem Namen verwaltet wird, können Sie sich ausschließlich auf die Anforderungen Ihrer Spiele und Backend-Dienste konzentrieren.

Sie können [AWS Lambda](#) verwenden, um Code auszuführen, ohne Server bereitstellen oder verwalten zu müssen. Lambda führt Ihren Code auf einer hochverfügbaren Recheninfrastruktur

aus und führt die Verwaltung der Rechenressourcen durch, einschließlich Server- und Betriebssystemwartung, Kapazitätsbereitstellung und automatische Skalierung sowie Protokollierung. Bei Lambda müssen Sie Ihren Code in einer der von Lambda unterstützten Sprachlaufzeiten bereitstellen. Lambda ist nützlich für die Verarbeitung von Spielereignissen, die Spielerauthentifizierung, die Verarbeitung von Käufen im Spiel und Matchmaking-Anfragen. Lambda skaliert automatisch auf der Grundlage der Anzahl der Ereignisse und kann unerwartete Verkehrsspitzen bewältigen.

[AWS Fargate](#) ist eine serverlose Compute-Engine für Container, die sowohl mit [Amazon Elastic Container Service](#) (ECS) als auch mit [Amazon Elastic Kubernetes Service](#) (EKS) funktioniert. AWS Fargate macht es einfach, sich auf die Entwicklung Ihrer Anwendungen zu konzentrieren, da die Bereitstellung und Verwaltung von Servern entfällt, Sie können Ressourcen pro Anwendung spezifizieren und bezahlen und verbessert die Sicherheit durch die Entwurfsisolierung von Anwendungen. Fargate ist ideal für Backend-Dienste, die Spielerprofile, Statusverwaltung und Matchmaking verwalten.

[Amazon GameLift](#) ist ein verwalteter Service für die Bereitstellung, den Betrieb und die Skalierung von dedizierten Spieleservern für sitzungsbasierte Multiplayer-Spiele. Sie können Ihren ersten Spieleserver in nur wenigen Minuten in der Cloud bereitstellen und so bis zu Tausende von Entwicklungsstunden für die Softwareentwicklung im Voraus sparen und die technischen Risiken verringern, die Entwickler häufig dazu veranlassen, Multiplayer-Funktionen aus ihren Designs zu streichen.

Implementierungsschritte

- Nutzen Sie AWS Lambda die automatische Skalierung und die serverlose Verwaltung für ereignisgesteuerte Workloads wie die Verarbeitung von Spielereignissen, Spielerauthentifizierung, In-Game-Käufen und Matchmaking-Anfragen.
- Verwenden Sie AWS Fargate ECS oder EKS für Backend-Dienste wie Spielerprofile, Statusverwaltung und Spielersuche, wodurch die Serververwaltung entfällt und die Anwendungsisolierung verbessert wird.
- Verwenden Sie Amazon GameLift , um dedizierte Spieleserver für sitzungsbasierte Multiplayer-Spiele bereitzustellen und zu skalieren und so die Entwicklungszeit und die betriebliche Komplexität zu reduzieren.

GAMESUS02-BP02 Passen Sie die richtige Größe Ihrer Rechenleistung an und setzen Sie GPU-Leistung nur dort ein, wo sie benötigt wird

Richten Sie Ihre Spieleserver und Ihr Backend so ein, dass Rechenressourcen effizient genutzt werden. Eine übermäßige Bereitstellung von Rechenleistung kann zu unnötigen Kosten führen und minimiert die Menge ungenutzter oder nicht ausgelasteter Ressourcen. GPU-Instances sollten verwendet werden, um spezifische Entwicklungsmaßnahmen wie HLOD-Rebuilds in Unreal zu unterstützen, oder wenn Ihre Spieleserver sie vom Design her erfordern. Dadurch werden die Umweltbelastung und die Kosten Ihrer Workloads erheblich reduziert.

Risikostufe, wenn diese bewährte Methode nicht eingeführt wird: Hoch

Implementierungsleitfaden

Sie sollten Ihre Spieleserver und Backend-Dienste so optimieren, dass sie mehrere EC2 Instanztypen und die geringste Anzahl benötigter Instanzen verwenden. Dadurch wird die Anzahl der verfügbaren Instanzen erhöht, um deinen Anforderungen während der Entwicklung oder beim Start deiner Spiele gerecht zu werden. Sie sollten den Instance-Typ auch an den spezifischen Workload anpassen, den Sie bereitstellen. Compute Optimized Instances unterstützen eine Vielzahl von Anwendungsfällen, darunter Spieleserver und Backend-Dienste wie Matchmaking. Speicheroptimierte Instances sind so konzipiert, dass sie eine schnelle Leistung für Workloads bieten, die große Datenmengen im Speicher verarbeiten. Verwenden Sie GPU-Instances nach Bedarf für hohe Leistungsanforderungen, jedoch nicht für allgemeine Rechenaufgaben. Wenn möglich, richten Sie Ihre Dienste oder Spieleserver so ein, dass sie mit [AWS Graviton-Instances auf ARM](#) laufen. Graviton ist der leistungsstärkste bis energieeffizienteste Instance-Typ, der auf dem Markt verfügbar ist. AWS Sie bieten im Vergleich zu x86-Instance-Typen auch eine verbesserte Leistung und höhere Kosten.

Verwenden Sie die, [AWS Compute Optimizer](#) um die optimalen AWS Ressourcenkonfigurationen zu identifizieren, z. B. Amazon Elastic Compute Cloud (EC2) -Instance-Typen, Amazon Elastic Block Store (EBS) -Volumenkonfigurationen, Aufgabengrößen der Amazon Elastic Container Service (ECS) -Services auf AWS Fargate, kommerzielle Softwarelizenzen, AWS Lambda Funktionsspeichergrößen und Amazon Relational Database Service (RDS) -DB-Instance-Klassen, und verwenden Sie maschinelles Lernen zur Analyse historischer Nutzungskennzahlen. Compute Optimizer bietet eine Reihe von Funktionen APIs und ein Konsolenerlebnis, um Kosten zu senken und die Workload-Leistung zu steigern, indem es die optimalen AWS Ressourcen für Ihre AWS Workloads empfiehlt.

Implementierungsschritte

- Ordnen Sie Rechenressourcen bestimmten Workloads zu, indem Sie rechneroptimierte Instanzen für Spieleserver, speicheroptimierte Instanzen für große Datenmengen und GPU-Instanzen nur für Aufgaben wie HLOD-Neuerstellungen oder GPU-abhängige Spieleserver verwenden.
- Optimieren Sie die Computernutzung, indem Sie nach Möglichkeit AWS Graviton-Instances einsetzen, um Energieeffizienz, bessere Leistung und Kosteneinsparungen im Vergleich zu x86-Instances zu erzielen.
- Wird verwendet AWS Compute Optimizer , um die historische Nutzung zu analysieren und die effizientesten Konfigurationen für AWS ECS EC2 - und Amazon RDS-Workloads zu empfehlen AWS Lambda, um Kosten zu senken und die Leistung zu verbessern.

Ressourcen

In den folgenden Ressourcen erfahren Sie mehr über unsere Best Practices im Bereich Nachhaltigkeit.

- [UN: Die Spielebranche rückt die Bedrohungen für den Planeten ins Rampenlicht](#)
- [Medium: Ökologische Nachhaltigkeit bei der Spieleentwicklung: Verantwortungsbewusstes Handeln für eine grünere Zukunft](#)

Die wichtigsten Dienstleistungen AWS

- [Amazon Aurora](#)
- [Amazon-DynamoDB](#)
- [Amazon DocumentDB \(mit MongoDB-Kompatibilität\)](#)
- [Amazon ElastiCache](#)
- [Amazon S3](#)
- [Amazon S3 S3-Speicherklassen](#)
- [Amazon S3 Intelligent-Tiering-Speicherklasse](#)
- [Amazon S3 Express One Zone-Speicherklasse](#)
- [Amazon Elastic Block Store](#)
- [AWS Lambda](#)
- [Amazon Elastic Container Service](#)

- [Amazon Elastic Kubernetes Service](#)
- [Amazon GameLift](#)
- [AWS Compute Optimizer](#)

Schlussfolgerung

Spiele sind darauf ausgelegt, einem weltweiten Spielerpublikum Unterhaltungserlebnisse zu bieten, und weisen Nutzungsmerkmale auf, die in der Regel unvorhersehbar und variabel sind. The Games Industry Lens beschreibt die häufigsten Arten von Szenarien, die typischerweise eine Spielarchitektur ausmachen, und bietet eine Reihe von Fragen und bewährten Methoden, die Sie bei der Entwicklung und dem Betrieb von Spielen in der Cloud berücksichtigen sollten. Indem Sie dieses Framework auf Ihre Spielarchitektur anwenden, können Sie zuverlässige, sichere, effiziente und kostengünstige Spiele in der Cloud entwickeln.

Mitwirkende

Die folgenden Personen haben zu diesem Dokument beigetragen:

- Adam Hatfield, leitender Lösungsarchitekt, Amazon Web Services
- Brady Webb, Technischer Kundenbetreuer, Amazon Web Services
- Bruce Ross — Senior Solutions Architect, Leiter von Well-Architected Lens, Amazon Web Services
- Caleb Cecil, stellvertretender Lösungsarchitekt, Amazon Web Services
- Carlos Perez, Cloud Optimization Success SA, Amazon Web Services
- Chase Herrington, Technischer Kundenbetreuer bei ESL, Amazon Web Services.
- Chris Blackwell, leitender Lösungsarchitekt, Amazon Web Services
- Corey Ouderkirk, Senior Technischer Kundenbetreuer, Amazon Web Services
- Derek Villavicencio, Prototyping SA, Amazon Web Services
- Erik Ynigo Becerril, leitender Lösungsarchitekt, Amazon Web Services
- Grzegorz Ochmanski, leitender Lösungsarchitekt, Amazon Web Services
- Hadrian Baron, Senior Technischer Kundenbetreuer, Amazon Web Services
- Ian Armbruster, Senior Customer Solutions Manager, Amazon Web Services
- Jed O Bray, leitender technischer Kundenbetreuer, Amazon Web Services
- Khurram Khokhar, Senior Technischer Kundenbetreuer, Amazon Web Services
- Kyle Somers, Senior Manager, Lösungsarchitektur, Amazon Web Services
- Madhuri Srinivasan, leitender technischer Redakteur, Well-Architected, Amazon Web Services
- Matthew Wygant, Senior TPM Guidance, Well-Architected, Amazon Web Services
- Nataliya Godunok, Cloud Optimization Success SA, Amazon Web Services
- Nirav Doshi, leitender Lösungsarchitekt, Amazon Web Services
- Olivia Liddell, Lösungsarchitektin, Amazon Web Services
- Randy James, Technischer Hauptkundenbetreuer, Amazon Web Services
- Reou Ando, Architekt für Spielösungen, Amazon Web Services
- Richard Raseley, Senior Technischer Kundenbetreuer, Amazon Web Services
- Sam Patzer, leitender Lösungsarchitekt, Amazon Web Services
- Scott Selinger, leitender Lösungsarchitekt, Amazon Web Services
- Sean Allen, Senior Solutions Architect, Amazon Web Services

- Serge Poueme, leitender Lösungsarchitekt, Amazon Web Services
- Stewart Matzek, leitender technischer Redakteur, Well-Architected, Amazon Web Services
- Trenton Potgieter, Senior Solutions Architect AI/ML/Analytics, Amazon Web Services

Dokumentversionen

Abonnieren Sie den RSS-Feed, um über Aktualisierungen des Whitepapers benachrichtigt zu werden.

Änderung	Beschreibung	Datum
Neue Objektivversion	Das gesamte Objektiv wurde mit neuen Best-Practice-Leitlinien aktualisiert.	9. Dezember 2025
Erste Veröffentlichung	Whitepaper erstmals veröffentlicht.	19. November 2021

AWS Glossar

Die neueste AWS Terminologie finden Sie im [AWS Glossar](#) in der AWS-Glossar Referenz.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.