AWS Whitepaper

Games Industry Lens

Games Industry Lens: AWS Whitepaper

Table of Contents

Riassunto e introduzione i
Disponibilità delle lenti 1
Principi di progettazione 3
Scenari 5
Hosting di giochi per un gameplay sincrono in tempo reale 5
Processi del server di gioco 6
Hosting di server di gioco basato su sessioni con backend senza server 8
Architettura multiregionale e ibrida per giochi a bassa latenza 10
Backend di gioco 11
Architettura di backend per giochi basata su container 12
Architettura di backend di gioco basata su server 14
Sviluppo di giochi su cloud (CGD) 17
Sviluppo di giochi su cloud: CI/CD 18
Sviluppo di giochi su cloud: workstation 20
Game Analytics Pipeline 21
Definizioni 24
Sistema di gioco 25
Server di gioco 26
Client di gioco 28
Messaggistica 28
Operazioni di gioco dal vivo (Live Ops) 30
Eccellenza operativa 31
Principi di progettazione 31
Operazioni dal vivo 32
GAMEOPS01-BP01 Utilizza gli obiettivi di gioco e le metriche delle prestazioni aziendali per sviluppare la tua strategia operativa dal vivo 33
Struttura dell'account 34
GAMEOPS02-BP01 Adotta una strategia multi-account per isolare giochi e applicazioni diversi nei propri account 34
GAMEOPS02-BP02 Organizza le risorse dell'infrastruttura utilizzando l'etichettatura delle risorse 38
Implementazioni di giochi 39
GAMEOPS03-BP01 Convalida e testa i sistemi e l'infrastruttura di gioco principali esistenti prima di riutilizzarli nel gioco 40

GAMEOPS03-BP02 Esegui la progettazione delle prestazioni prima di ogni versione (o almeno per le versioni principali) 41
GAMEOPS03-BP03 Test di caricamento precoce e frequente 42
GAMEOPS03-BP04 Adotta una strategia di distribuzione che riduca al minimo l'impatto sui giocatori 44
GAMEOPS03-BP05 È necessaria un'infrastruttura pre-scalabile per supportare i requisiti di picco 48
Monitoraggio della salute 50
GAMESOPS04-BP01 Strumenta il gioco per rilevare e monitorare i problemi che influiscono sui giocatori 51
Test di caricamento 52
GAMEOPS05-BP01 Scegli lo stage, l'architettura e il framework di test di carico giusti per raggiungere i tuoi obiettivi 52
Ottimizzazione nel tempo 56
GAMEOPS06-BP01 Monitora le metriche chiave del gioco per identificare le tendenze e i modelli dei giocatori e utilizza le informazioni per migliorare il gioco 56
GAMEOPS06-BP02 Aggiorna e adatta l'approccio al test di carico man mano che il gioco cambia 58
Resources 59
Documentazione e blog 59
Soluzioni dei partner 60
Whitepaper 61
Video 61
Materiali per la formazione 61
Sicurezza 62
Principi di progettazione 63
Nozioni di base sulla sicurezza 63
GAMESEC01-BP01 Usa i ruoli e l'accesso federato, anziché l'utente root dell'account, per eseguire azioni sul tuo ambiente AWS 64
GAMESEC01-BP02 Utilizzato per configurare rapidamente un ambiente multi-account su AWS Control Tower AWS 65
GAMESEC01-BP03 Utilizza politiche di ruolo con privilegi minimi adattate a funzioni lavorative specifiche 67
GAMESEC01-BP04 Usa ruoli e politiche di accesso federato insieme a politiche di accesso a livello di account per concedere l'accesso alle tue risorse AWS 67
GAMESEC01-BP05 Usa un provider di identità centrale 69

Sicurezza continua 70
GAMESEC02-BP01 Usa modelli pronti per l'implementazione per pratiche di sicurezza standard 70
GAMESEC02-BP02 Usa tecniche di riparazione automatizzate quando si verifica un evento di sicurezza 71
Gestione dell’identità e degli accessi 73
GAMESEC03-BP01 Determina il tuo approccio per identificare e controllare l'accesso dei giocatori all'ambiente e alle risorse del tuo gioco 73
GAMESEC03-BP02 Autentica le richieste inviate al tuo servizio di backend di gioco 75
GAMESEC03-BP03 Usa il servizio di backend di gioco per convalidare le richieste dei giocatori di partecipare a una partita multiplayer 77
GAMESEC03-BP04 Applica una rigorosa politica di sicurezza per gli account utente dei giocatori richiedendo una password complessa 78
GAMESEC03-BP05 Fornisci ai giocatori la possibilità di configurare l'autenticazione a più fattori (MFA) sui propri account 79
Controllo accessi 80
GAMESEC04-BP01 Limita l'accesso ai contenuti scaricabili a client e utenti autorizzati 80
GAMESEC04-BP02 Limita l'accesso all'origine alle reti di distribuzione dei contenuti autorizzate () CDNs 83
GAMESEC04-BP03 Implementa restrizioni geografiche per limitare l'accesso non autorizzato 84
GAMESEC04-BP04 Limita l'accesso ai contenuti con soluzioni di gestione dei diritti digitali (DRM) 85
Rilevamento 86
GAMESEC05-BP01 Implementa una strategia completa di raccolta dati per monitorare il comportamento dei giocatori 87
GAMESEC05-BP02 Raccogli, archivia e analizza i registri di utilizzo dei giocatori per rilevare comportamenti inappropriati 88
Protezione dell'infrastruttura 89
GAMESEC06-BP01 Usa strumenti per rilevare e rispondere alle minacce alla tua infrastruttura 89
GAMESEC06-BP02 Usa l'intelligenza artificiale e gli strumenti di apprendimento automatico per automatizzare alcuni aspetti della tua strategia di protezione dell'infrastruttura 91
GAMESEC06-BP03 Usa le informazioni ricavate dai log a livello di sistema per migliorare continuamente la tua strategia di protezione dell'infrastruttura 92
Risposta agli incidenti 93

GAMESEC07-BP01 Implementa un piano di risposta agli incidenti per gestire i cattivi attori e i comportamenti abusivi 94
GAMESEC07-BP02 Bandisci gli account associati a malintenzionati 94
Sicurezza delle applicazioni 95
GAMESEC08-BP01 Applica la sicurezza in ogni fase della pipeline CI/CD 96
Automatizza la sicurezza 97
GAMESEC09-BP01 Integra strumenti e automazione per ridurre il tempo medio delle revisioni di sicurezza 98
Modellazione delle minacce 99
GAMESEC10-BP01 Determina quando e come completare gli esercizi di modellazione delle minacce durante l'intero ciclo di vita dello sviluppo delle applicazioni 99
Resources 101
Affidabilità 102
Principi di progettazione 102
Fondamenti 103
Architettura del carico di lavoro 103
GAMEREL01-BP01 Distribuisci l'infrastruttura di gioco su più zone e regioni di disponibilità per migliorare la resilienza 103
Gestione delle modifiche 105
GAMEREL02-BP01 Implementa una strategia di scalabilità che incorpori lo stato delle sessioni di gioco attive dei giocatori 106
GAMEREL02-BP02 Supporta l'uso di più tipi di istanze per il tuo gioco EC2 107
Gestione dei guasti 108
GAMEREL03-BP01 Monitora le interruzioni dei server di gioco e utilizza i dati per migliorare l'architettura di hosting per raggiungere gli obiettivi di affidabilità 109
GAMEREL03-BP02 Implementa una combinazione libera delle funzionalità di gioco per gestire gli errori con un impatto minimo sull'esperienza del giocatore 110
GAMEREL03-BP03 Monitora gli eventi dell'infrastruttura nel tempo per misurare l'impatto sul comportamento dei giocatori 112
Resources 114
Efficienza delle prestazioni 116
Principi di progettazione 116
Scelta dell'architettura 117
GAMEPERF01-BP01 Valuta i requisiti di risorse e le esigenze di scalabilità dei server di gioco 118

GAMEPERF01-BP02 Considera il sovraccarico operativo per il ridimensionamento dei server di gioco 119
GAMEPERF01-BP03 Valuta l'integrazione con altri servizi, ambienti di sviluppo, architetture CPU target e funzionalità AWS 120
Selezione della regione 122
GAMEPERF02-BP01 Seleziona una regione d'origine vicina ai tuoi giocatori 122
GAMEPERF02-BP02 Progetta un approccio che supporti la collocazione di infrastrutture di gioco sensibili alla latenza vicino ai giocatori per migliorare le prestazioni 123
Sviluppo iterativo 126
GAMEPERF03-BP01 Usa Amazon GameLift Anywhere e un toolkit di test GameLift 126
GAMEPERF03-BP02 Verifica le prestazioni e la scalabilità dei server di gioco 128
GAMEPERF03-BP03 Ottimizza l'utilizzo delle risorse dei GameLift contenitori 129
Calcolo e hardware 130
GAMEPERF04-BP01 Monitora i processi del server di gioco per rilevare problemi 130
GAMEPERF04-BP02 Metti alla prova le prestazioni del tuo server di gioco con scenari di gioco simulati e reali 132
Calcola la selezione 133
GAMEPERF05-BP01 Valuta le tue prestazioni di gioco su più tipi di elaborazione 133
GAMEPERF05-BP02 Sposta le attività di non-latency-sensitive calcolo in flussi di lavoro asincroni 135
Gestione dei dati 136
GAMEPERF06-BP01 Centralizza la raccolta e l'archiviazione dei log 137
GAMEPERF06-BP02 Categorizza e archivia i dati di gioco in base ai modelli di accesso 137
GAMEPERF06-BP03 Abilita la formattazione e il raggruppamento efficienti dei log 138
GAMEPERF06-BP04 Implementa politiche di rotazione e conservazione dei log 139
GAMEPERF06-BP05 Utilizza strumenti di monitoraggio e visualizzazione 139
Reti e distribuzione di contenuti 139
GAMEPERF07-BP01 Definisci le soglie di latenza di rete per il tuo gioco 140
GAMEPERF07-BP02 Esegui un servizio di matchmaking separato per ogni modalità di gioco e regione di hosting del gioco 140
GAMEPERF07-BP03 Monitora regolarmente le prestazioni del matchmaking 141
GAMEPERF07-BP04 Monitora regolarmente le prestazioni di rete 142
GAMEPERF07-BP05 Utilizza la tecnologia di accelerazione della rete per migliorare le prestazioni su Internet 143
Processo e cultura 145
GAMEPERF08-BP01 Informa e includi il giocatore nel processo 145

GAMEPERF08-BP02 Allinea la selezione della soluzione alle capacità e all'esperienza del team di progettazione 147
Resources 147
Ottimizzazione dei costi 150
Principi di progettazione 151
Implementazione della gestione finanziaria del cloud 152
Comprensione delle spese e dell'utilizzo 152
GAMECOST01-BP01 Implementa l'attribuzione del costo per giocatore, funzionalità di gioco e ambiente 152
GAMECOST01-BP02 Scopri le opportunità di ottimizzazione 154
Risorse convenienti in termini di costo 155
GAMECOST02-BP01 Ottimizza il costo del trasferimento dei dati su Internet 156
GAMECOST02-BP02 Ottimizza il numero di sessioni di gioco ospitate su ciascuna istanza del server di gioco per ottimizzare i costi 157
GAMECOST02-BP03 Seleziona l'opzione di determinazione dei prezzi di elaborazione appropriata per ridurre i costi 158
Costi per il trasferimento dati 161
GAMECOST03-BP01 Scegli il tipo di archiviazione appropriato per i contenuti generati dagli utenti per ridurre i costi 162
GAMECOST03-BP02 Ottimizza i database per i backend di gioco 163
Gestione delle risorse relative alla domanda e all'offerta 165
Ottimizzazione nel tempo 165
Resources 165
Sostenibilità 167
Principi di progettazione 167
Selezione della regione 168
Allineamento alla domanda 168
Software e architettura 168
Gestione dei dati 168
GAMESUS01-BP01 Utilizza tecnologie di archiviazione che si adattano ai modelli adattati ai contenuti degli utenti, alle informazioni sugli abbonati e agli acquisti in-game 168
GAMESUS01-BP02 Utilizza le politiche del ciclo di vita o la scadenza TTL per eliminare giochi, dati utente, file di registro o risorse obsolete non necessari 171
Hardware e servizi 173
GAMESUS02-BP01 Seleziona i servizi gestiti per carichi di lavoro di elaborazione appropriati 174

GAMESUS02-BP02 Dimensiona correttamente il tuo calcolo e implementa le prestazioni della GPU solo dove necessario 175
Resources 176
AWS Servizi chiave 176
Conclusioni 178
Collaboratori 179
Revisioni del documento 181
AWS Glossario 182
..... clxxxiii

Games Industry Lens — AWS Well-Architected Framework

Data di pubblicazione: 9 dicembre 2025 () Revisioni del documento

AWS Well-Architected Framework aiuta gli architetti del cloud a creare un'infrastruttura sicura, ad alte prestazioni, resiliente ed efficiente per le loro applicazioni e carichi di lavoro. Basato su sei pilastri: eccellenza operativa, sicurezza, affidabilità, efficienza delle prestazioni, ottimizzazione dei costi e sostenibilità, Well-Architected offre a clienti e partner un approccio coerente per valutare le architetture, correggere i rischi AWS e implementare progetti che offrano valore aziendale.

In quest'ottica, ci concentriamo su come progettare, implementare e progettare i carichi di lavoro dei giochi nel. Cloud AWS Definiamo i componenti, esploriamo scenari di carico di lavoro comuni e delineiamo i principi di progettazione che aiutano ad applicare il Well-Architected Framework. Ti consigliamo di iniziare a progettare la tua architettura prendendo in considerazione le best practice e le domande del white paper AWS Well-Architected Framework. Questo documento fornisce best practice supplementari per i clienti del settore dei giochi.

Questo obiettivo specifica le migliori pratiche che hanno lo scopo di affrontare le caratteristiche uniche della creazione e della gestione di giochi nel cloud, sulla base della nostra esperienza di collaborazione con sviluppatori ed editori del settore dei giochi in tutto il mondo. Fornisce indicazioni su come progettare e gestire l'ambiente in modo che sia ottimizzato in termini di costi e scalabile per far fronte alle fluttuazioni della domanda globale degli operatori. Questo obiettivo fornisce anche indicazioni per proteggere l'infrastruttura di gioco e ottimizzare le prestazioni per offrire un'esperienza di gioco positiva.

Questo documento è destinato a coloro che ricoprono ruoli tecnologici, come ad esempio Chief Technology Officer (CTOs), direttori tecnici degli studi di gioco, architetti, sviluppatori e membri del team operativo. Dopo aver letto questo documento, comprenderai le AWS migliori pratiche e le strategie da utilizzare nella progettazione di architetture per i giochi.

Disponibilità delle lenti

Le lenti personalizzate ampliano le linee guida sulle migliori pratiche fornite da AWS Well-Architected Tool. AWS WA Tool ti consente di creare lenti personalizzate o di utilizzare lenti create da altri che sono state condivise con te.

Per iniziare a esaminare il carico di lavoro dei tuoi giochi, scarica e importa Games Industry Lens AWS Well-Architected Tool dall'archivio pubblico di lenti personalizzate AWS Well-Architected. GitHub

Principi di progettazione

Il AWS Well-Architected Framework identifica i seguenti principi generali di progettazione per facilitare una buona progettazione nel cloud per i carichi di lavoro dei giochi:

- Comprendi il comportamento e i modelli di utilizzo dei giocatori per far evolvere il gioco e contribuire a proteggerli: per favorire il miglioramento continuo del gioco e gestire efficacemente l'esperienza del giocatore, è importante avere visibilità sul modo in cui i giocatori interagiscono con il gioco stesso e con il resto dei giocatori. Questo ti aiuta a capire come migliorare il gioco, gestire i costi, monitorare e reagire alle attività di utilizzo non autorizzate che mettono a rischio l'esperienza del giocatore.
- Utilizza tecnologie che semplificano le operazioni di gioco e aumentano la velocità di sviluppo: dai la priorità all'adozione di tecnologie in grado di migliorare la velocità e ridurre il sovraccarico operativo legato all'offerta di nuove funzionalità e miglioramenti ai giocatori. I giochi sono incentrati sui successi e i giocatori possono prendere in considerazione molte scelte, quindi muoversi rapidamente e adattarsi ai cambiamenti sono fondamentali per il successo di un gioco. Valuta se ti senti a tuo agio con il tuo software o se preferisci adottare un servizio gestito analogo offerto da AWS, AWS Partner o entrambi.
- Ottimizza la tua architettura per migliorare le metriche che riflettono l'esperienza effettiva del giocatore: man mano che adatti ed evolvi la tua architettura nel tempo, pensa a come questi miglioramenti e modifiche influiranno sull'esperienza del giocatore. I carichi di lavoro dei giochi dovrebbero essere in grado di resistere e ridurre al minimo l'impatto dei guasti per bloccare interruzioni diffuse del gioco. Le funzionalità e i sistemi di gioco che non dipendono in modo critico l'uno dall'altro dovrebbero essere disaccoppiati per ridurre l'impatto degli errori e isolare i problemi che influiscono sui giocatori.
- Progetta l'infrastruttura per soddisfare il picco di concorrenza tra giocatori e scalala dinamicamente in base alle esigenze: l'infrastruttura deve essere progettata per soddisfare la domanda dei giocatori. Le metriche, come la concomitanza delle sessioni dei giocatori e il numero di accessi, possono essere utilizzate per scalare preventivamente prima che i sistemi si sovraccarichino. Le metriche di utilizzo reattivo del sistema, come il consumo di CPU e memoria, possono essere utilizzate per scalare dopo il sovraccarico dei sistemi. Scalando dinamicamente l'infrastruttura, puoi ridurre i costi operativi del gioco.
- Implementa i runbook per migliorare le operazioni di gioco: i runbook operativi devono essere utilizzati per gestire in modo coerente le attività ricorrenti delle operazioni di gioco. I runbook dovrebbero essere utilizzati per i flussi di lavoro più comuni relativi alle operazioni di gioco,

come l'analisi e la risposta alle segnalazioni dei giocatori, la gestione delle attività di pre-dimensionamento dell'infrastruttura per prepararsi a eventi su larga scala previsti, come il lancio di nuove stagioni e il rilascio di contenuti di gioco, e per gestire le tipiche attività di manutenzione del gioco.

Scenari

In questa sezione, trattiamo diversi scenari comuni in un'architettura di gioco. Ogni scenario include le caratteristiche comuni che guidano la progettazione e un esempio di diagramma di architettura di riferimento.

Scenari

- Hosting di giochi per un gameplay sincrono in tempo reale
- Backend di gioco
- Architettura di backend di gioco basata su server
- Sviluppo di giochi su cloud (CGD)
- Game Analytics Pipeline

Hosting di giochi per un gameplay sincrono in tempo reale

Il gameplay sincrono in tempo reale consente a due o più giocatori di partecipare e interagire contemporaneamente a una partita, condividendo lo stato del gameplay tra i giocatori connessi per creare un'esperienza il più possibile simile a quella in tempo reale. Esempi di giochi sincroni includono sparatutto in prima persona, giochi online multigiocatore di massa (MMOG), giochi sportivi e d'azione o un gioco online in cui due o più giocatori devono essere collegati per condividere l'esperienza di gioco quasi in tempo reale.

Le caratteristiche delle architetture di gioco sincrone in tempo reale includono:

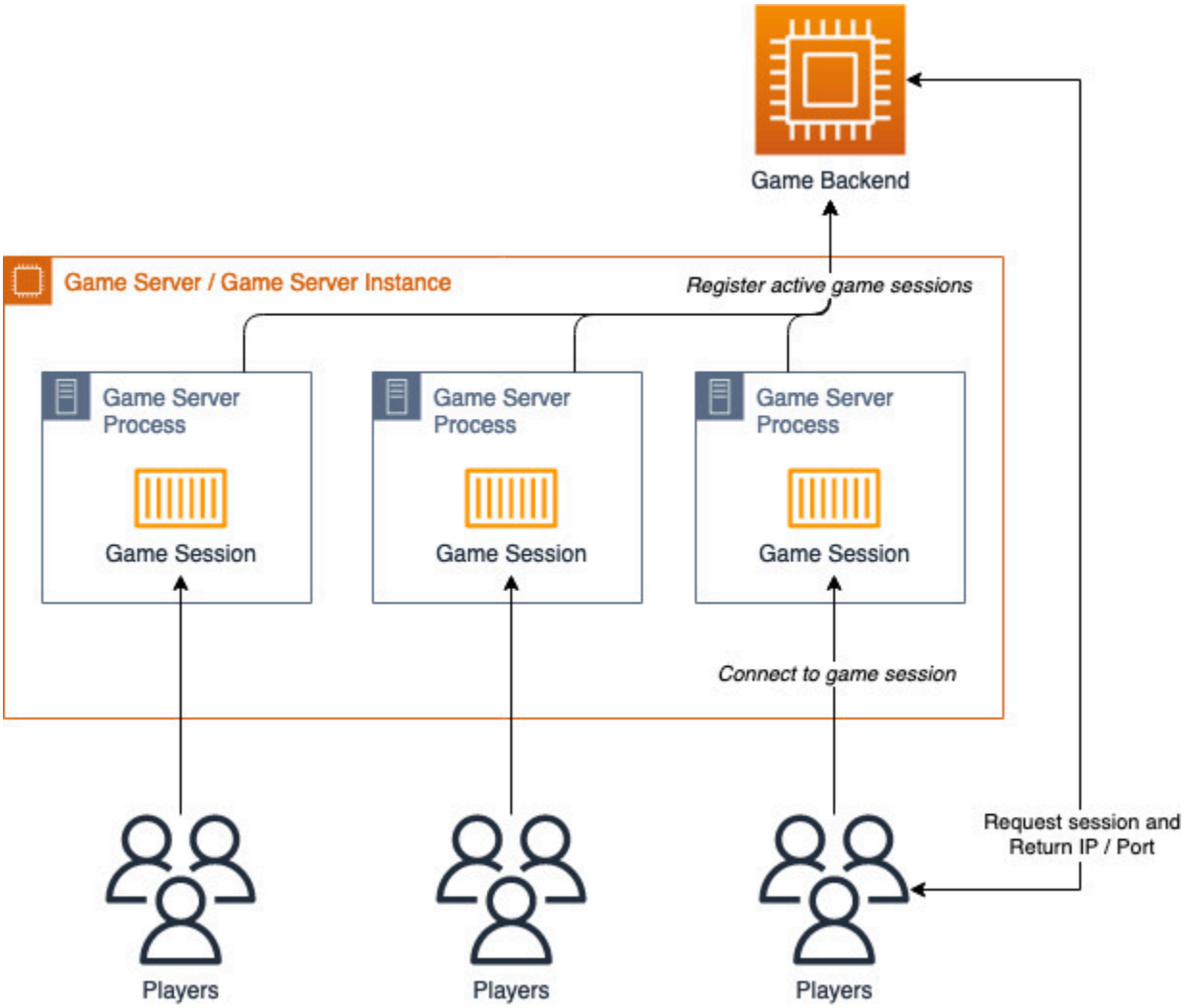
- Alcuni giochi possono essere ospitati come sessioni di gioco tramite processi di server di gioco eseguiti su server dedicati. Alcuni giochi possono utilizzare architetture simili al P2P che utilizzano server Session Traversal Utilities for NAT (STUN) o Traversal Using Relays around NAT (TURN) più leggere. Indipendentemente dal tipo di server utilizzato, i server di gioco sono ospitati in più data center e in tutto il mondo. Regioni AWS
- I client di gioco possono partecipare a una sessione di gioco richiedendo una partita a un servizio di matchmaking centralizzato ospitato nel sistema di backend del gioco o selezionando una partita da un elenco predefinito di server di gioco disponibili. Il client di gioco è dotato di un indirizzo IP e di una porta a cui connettersi.
- Molti giochi sincroni sono sensibili alla latenza, come gli sparatutto in prima persona e i giochi online multigiocatore di massa. Probabilmente includeranno algoritmi come il rewind e la

dilatazione temporale per ridurre al minimo gli effetti del ritardo, ma potrebbero anche avere una tolleranza di latenza predefinita che viene accuratamente misurata e ottimizzata per ridurre l'esperienza di ritardo che a volte può verificarsi per i giocatori in situazioni di alta latenza. Queste informazioni sulla latenza vengono determinate utilizzando i client di gioco per eseguire il ping del server di gioco disponibile per acquisire metriche come latenza, jitter di rete e altre metriche importanti Regioni AWS per l'esperienza di gioco. Queste metriche vengono inviate a un servizio centrale di raccolta delle metriche nel sistema di backend di gioco, in modo che gli stream operativi in diretta possano monitorare lo stato del gioco. Durante il processo di matchmaking, i client di gioco forniscono i dati di latenza correnti come uno dei parametri di richiesta quando richiedono una partita, e il servizio di matchmaking può utilizzare tali dati di latenza come una delle variabili quando seleziona un server di gioco per ospitare il giocatore.

- In genere, il gioco si svolge su una combinazione di protocolli (ad esempio server di gioco che utilizzano una messaggistica basata su UDP più veloce con matchmaking, autenticazione e altro traffico client-server tramite HTTPS).
- I server di gioco utilizzano algoritmi e design per ridurre al minimo il traffico client-server, come lo streaming di trasformazione, i delta e la compressione dei dati.
- I server di gioco sono spesso bersagli di attività dannose e dovrebbero essere protetti con una soluzione di protezione S come. DDo AWS Shield Advanced

Processi del server di gioco

Il diagramma seguente illustra una tipica architettura di server di gioco. Descrive la relazione logica tra un'istanza del server di gioco e i processi del server di gioco che ospitano le sessioni di gioco.



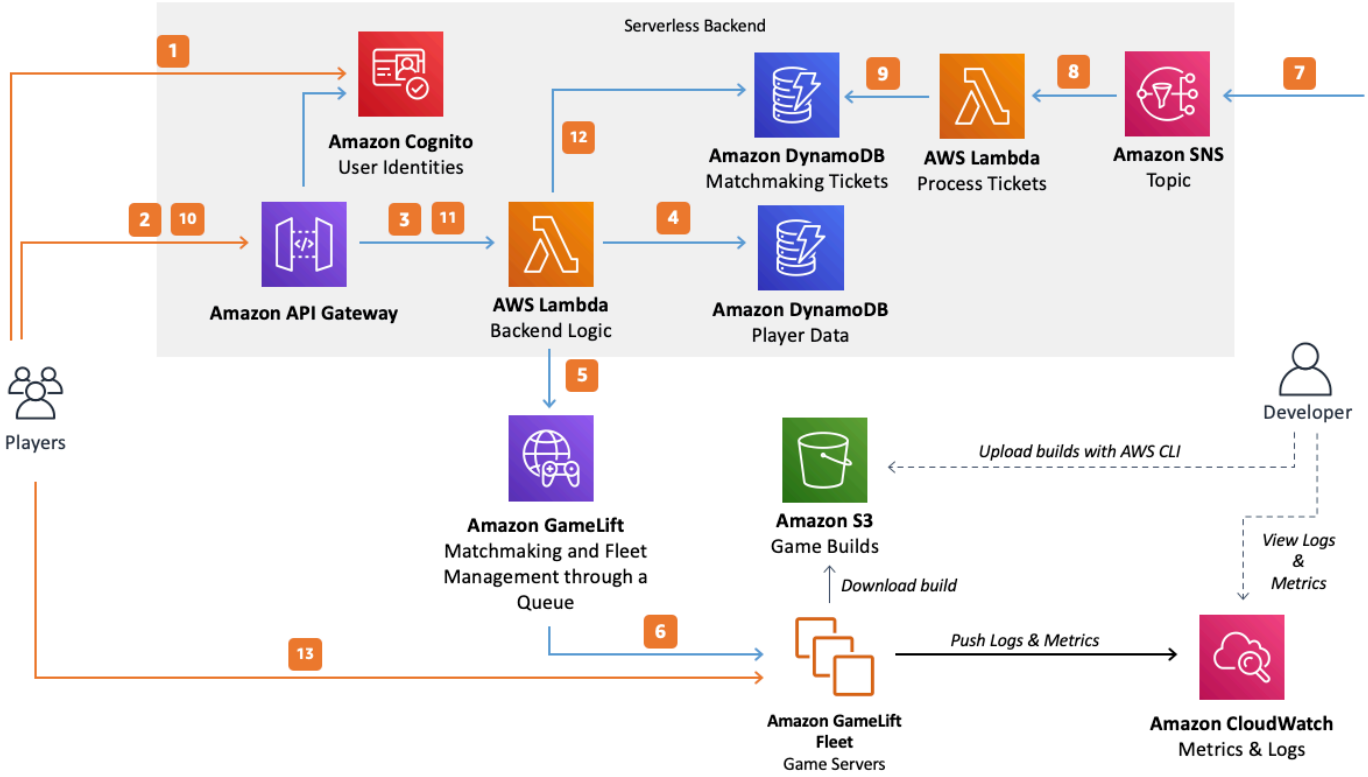
Architettura logica del server di gioco

- Un' EC2 istanza Amazon viene utilizzata come server di gioco, nota anche come istanza di server di gioco. Il server di gioco ospita uno o più processi del server di gioco, ognuno dei quali esegue una copia della build del server di gioco. In genere, su un'istanza del server di gioco vengono eseguiti più processi del server di gioco per utilizzare le risorse di elaborazione in modo efficiente e ridurre i costi. Quando una sessione di gioco è attiva e pronta per ospitare sessioni di giocatori, il suo stato viene aggiornato con il backend di gioco (di solito un servizio di matchmaking) in modo che possa iniziare a essere utilizzata per ospitare giocatori.
- Il backend di gioco può fornire al giocatore l'indirizzo IP e la porta del server che ospita una sessione di gioco in modo che possa connettersi per giocare.

Hosting di server di gioco basato sulla sessione con backend senza server

Quando sviluppi un'architettura per il tuo gioco, considera le caratteristiche e le capacità di cui hai bisogno e il livello di gestione operativa che sei disposto a gestire. Per offrire il miglior equilibrio tra facilità d'uso e flessibilità, puoi creare il tuo gioco utilizzando i servizi gestiti dei provider di cloud. I servizi gestiti ti offrono il controllo necessario per sviluppare e personalizzare le tue funzionalità di gioco personalizzate, riducendo al contempo il carico di implementazione e gestione dell'infrastruttura.

Per ospitare un gioco multiplayer basato su sessioni è necessario disporre di un'infrastruttura server per ospitare i processi del server di gioco e di un backend scalabile per il matchmaking e la gestione delle sessioni. La seguente architettura di riferimento mostra come utilizzare l'hosting GameLift gestito di Amazon e un backend serverless per gestire i giochi basati su sessioni.



Hosting GameLift gestito da Amazon per giochi basati su sessioni

Il diagramma descrive il processo per far sì che i giocatori accedano ai giochi in esecuzione su un hosting di giochi GameLift gestito. Include i seguenti passaggi:

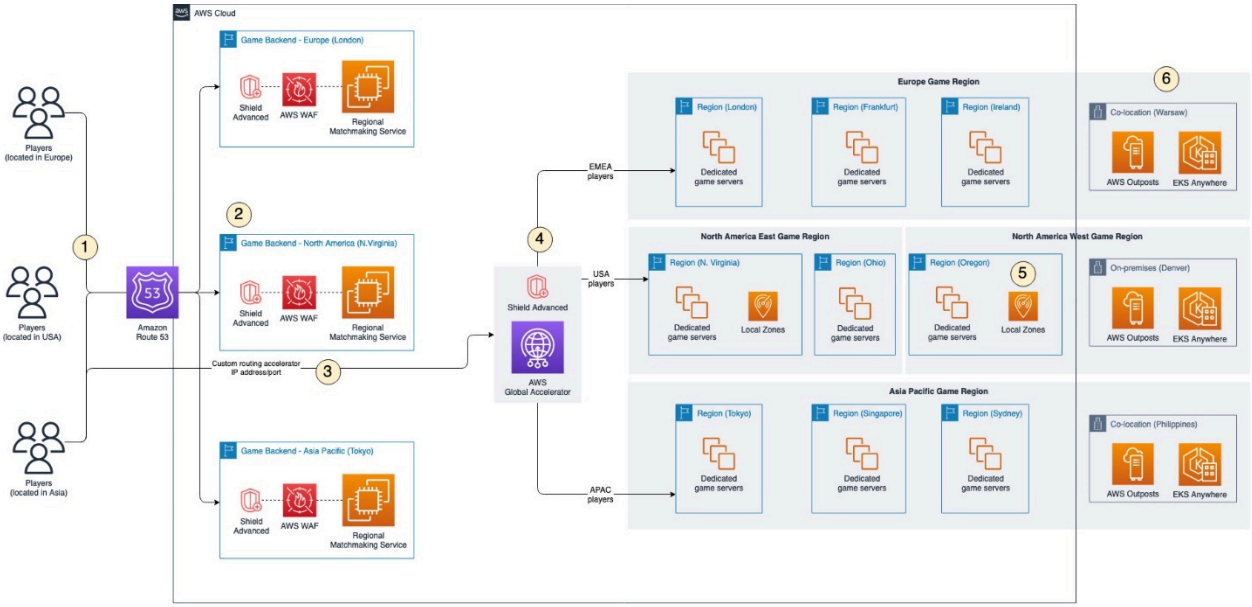
1. Il client di gioco richiede un'identità Amazon Cognito da un pool di identità Amazon Cognito. Questo può essere collegato facoltativamente a provider di identità esterni.

2. Il client di gioco riceve credenziali di accesso temporanee e richiede una sessione di gioco tramite un Amazon API Gateway firmando la richiesta con le credenziali di Amazon Cognito.
3. API Gateway richiama una AWS Lambda funzione.
4. La funzione Lambda richiede i dati dei giocatori da una tabella Amazon DynamoDB. L'identità di Amazon Cognito viene utilizzata per richiedere in modo sicuro i dati corretti del giocatore perché l'identità autenticata viene fornita nei dati contestuali della richiesta.
5. Utilizzando i dati corretti del giocatore per ulteriori informazioni (come il livello di abilità del giocatore), la funzione Lambda richiede una partita tramite GameLift FlexMatch matchmaking. Puoi definire una configurazione di FlexMatch matchmaking con documenti di configurazione basati su JSON. Il client di gioco può generare metriche di latenza eseguendo il ping degli endpoint del server in varie regioni, e i dati sulla latenza possono essere utilizzati per supportare il matchmaking basato sulla latenza.
6. Dopo aver FlexMatch abbinato un gruppo adeguato di giocatori con una latenza adeguata in una regione, richiede che la sessione di gioco venga posizionata in coda. GameLift La coda contiene flotte con una o più sedi regionali registrate.
7. Quando la sessione viene effettuata in una delle sedi della flotta, viene inviata una notifica di evento a un argomento di Amazon SNS.
8. Una funzione Lambda riceverà l'evento Amazon SNS e lo elaborerà.
9. Se il messaggio Amazon SNS è un MatchmakingSucceeded evento, la funzione Lambda scrive il risultato su DynamoDB con la porta del server e l'indirizzo IP. Un valore time-to-live (TTL) viene utilizzato per garantire che i ticket di matchmaking vengano eliminati da DynamoDB quando non sono più necessari.
10. Il client di gioco invia una richiesta firmata ad API Gateway per verificare lo stato del ticket di matchmaking in un intervallo specifico.
11. API Gateway richiama una funzione Lambda che verifica lo stato del ticket di matchmaking.
12. La funzione Lambda controlla DynamoDB per determinare se il ticket ha avuto successo. Se ha avuto successo, la funzione Lambda invia l'indirizzo IP, la porta e l'ID della sessione del giocatore al client. Se il ticket non è riuscito, la funzione Lambda invia una risposta dichiarando che la partita non è pronta.
13. Il client di gioco si connette al server di gioco tramite TCP o UDP utilizzando la porta e l'indirizzo IP forniti dal backend. Invia l'ID della sessione del giocatore al server di gioco e il server di gioco lo convalida utilizzando l'SDK Amazon GameLift Server.

In alternativa, puoi modificare l'architettura precedente per utilizzare API Gateway WebSockets con Amazon GameLift. In questo approccio, la comunicazione tra il client di gioco e il servizio di backend di gioco avviene tramite un'implementazione WebSocketbasata. Questa implementazione può essere utilizzata in modo che la funzione Lambda del backend del gioco invii un messaggio lato server al client di gioco utilizzando WebSocket un modello di polling anziché implementare.

Architettura multiregionale e ibrida per giochi a bassa latenza

Questa sezione descrive un'architettura multiregionale e ibrida per giochi a bassa latenza.



Riduzione della latenza con l'accelerazione della rete e i server di gioco distribuiti a livello globale

1. I giocatori di un gioco disponibile a livello globale possono provenire da qualsiasi luogo. Quando un giocatore richiede una sessione di gioco o una partita, il suo client di gioco invia una richiesta al servizio di backend di gioco registrato con Amazon Route 53. Il routing basato sulla latenza di Route 53 può essere utilizzato per indirizzare il giocatore al backend di gioco disponibile più vicino.
2. Il backend di gioco viene utilizzato in diverse popolazioni di giocatori più vicine. Regioni AWS Ogni backend di gioco include un servizio di matchmaking regionale che trova una sessione di gioco da tutte le regioni di gioco. Sebbene la richiesta di matchmaking di un giocatore venga elaborata da un servizio di matchmaking regionale vicino a lui, il servizio di matchmaking è in grado di indirizzare un giocatore a una sessione di gioco in un'altra regione di gioco, se necessario. Questa azione migliora la resilienza e le prestazioni. Inoltre, ogni servizio di backend di gioco utilizza AWS WAF e AWS Shield Advanced fornisce filtri web di livello 7, controllo dei bot e protezioni contro gli attacchi Distribution Denial of Service (S). DDo Hai molte opzioni per creare il tuo servizio di

backend di gioco, ad esempio serverless, container, EC2 istanze o hosting del servizio di backend di gioco nei tuoi data center.

3. Per migliorare l'esperienza dei giocatori riducendo la latenza e il jitter della rete, viene implementato un acceleratore di routing personalizzato utilizzando AWS Global Accelerator, che ottimizza automaticamente il routing del traffico dal client di gioco al server di gioco utilizzando la rete globale. AWS Puoi configurare l'acceleratore di routing personalizzato per mappare le porte del listener Global Accelerator alla porta dell'istanza del tuo server di gioco. EC2 Il client di gioco si connette all'IP e alla porta Global Accelerator che funge da proxy e indirizza i giocatori in modo deterministico all'IP e alla porta corretti del server di gioco che ospitano la sessione di gioco.
4. Il tuo gioco include regioni di gioco logiche adatte ai giocatori che rappresentano un insieme di località che ospitano server di gioco geograficamente vicine tra loro, come il Nord America o l'Asia Pacifico. Per ridurre la latenza e aumentare la copertura geografica, è possibile utilizzare una combinazione di diverse soluzioni di hosting di server di gioco per migliorare l'esperienza del giocatore. Dai la priorità all'utilizzo di Regioni AWS laddove possibile, poiché queste postazioni sono dotate di tutte le funzionalità e offrono il massimo ingombro in termini di capacità.
5. Usa AWS Local Zones per ospitare server di gioco in aree geografiche di giocatori con scarsità di servizi in cui non disponi di strutture di hosting esistenti o non Regione AWS sono disponibili.
6. AWS Outposts Implementalo nei data center locali e nei provider di co-location esistenti per creare un piano di controllo e un'esperienza di gestione senza interruzioni in ogni sede di implementazione utilizzando rack completamente gestiti e server montati su rack. AWS Outposts sono utili anche se non disponi della capacità del server esistente nell'ambiente locale. Tuttavia, se desideri un'implementazione ibrida con software in esecuzione sulla tua infrastruttura server esistente, dovresti usare Amazon EKS Anywhere, che ti consente di creare ed eseguire cluster Kubernetes sulla tua infrastruttura con connettività ad Amazon EKS. Ciò fornisce una visualizzazione coerente da console dei cluster Kubernetes in locale.

Backend di gioco

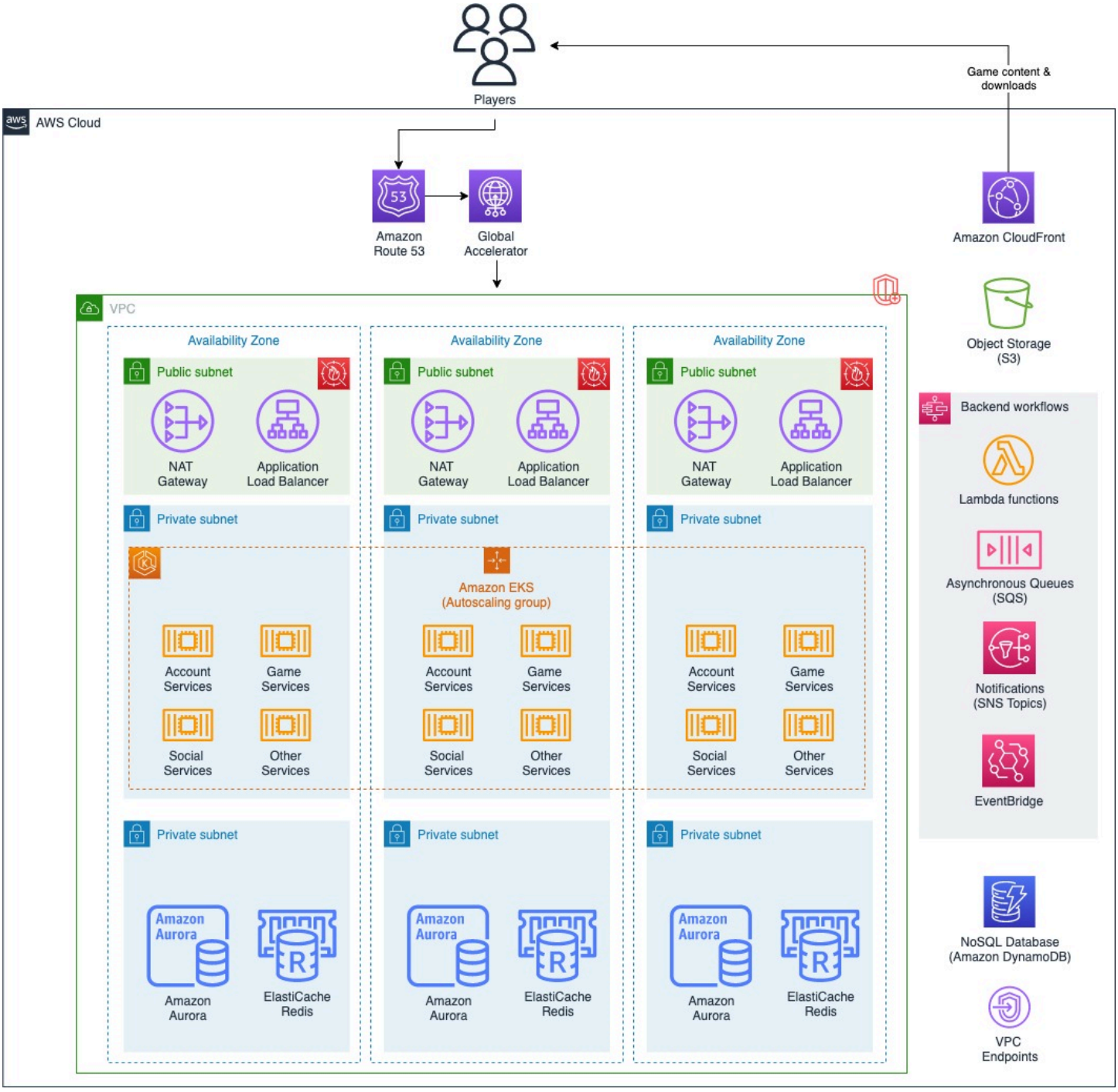
I backend di gioco vengono utilizzati per gestire il gioco e lo stato del giocatore, nonché per integrare nel gioco funzionalità social e a livello di sistema che supportano l'esperienza di gioco. La gestione dei profili dei giocatori, la memorizzazione degli oggetti e dell'inventario, le statistiche e le classiifiche sono esempi di servizi ospitati nei backend di gioco.

I backend di gioco sono in genere creati come REST a cui i APIs client accedono tramite HTTPS. Tuttavia, sono comuni anche altri approcci, come WebSockets quello di fornire canali bidirezionali per casi d'uso, come le notifiche dei client per la chat e la presenza all'interno del gioco. I backend

di gioco possono essere implementati utilizzando una varietà di architetture di distribuzione diverse, incluso l'utilizzo di istanze, contenitori o un'architettura serverless.

Architettura di backend di gioco basata su container

Questa sezione descrive un'architettura di backend di gioco basata su contenitori.



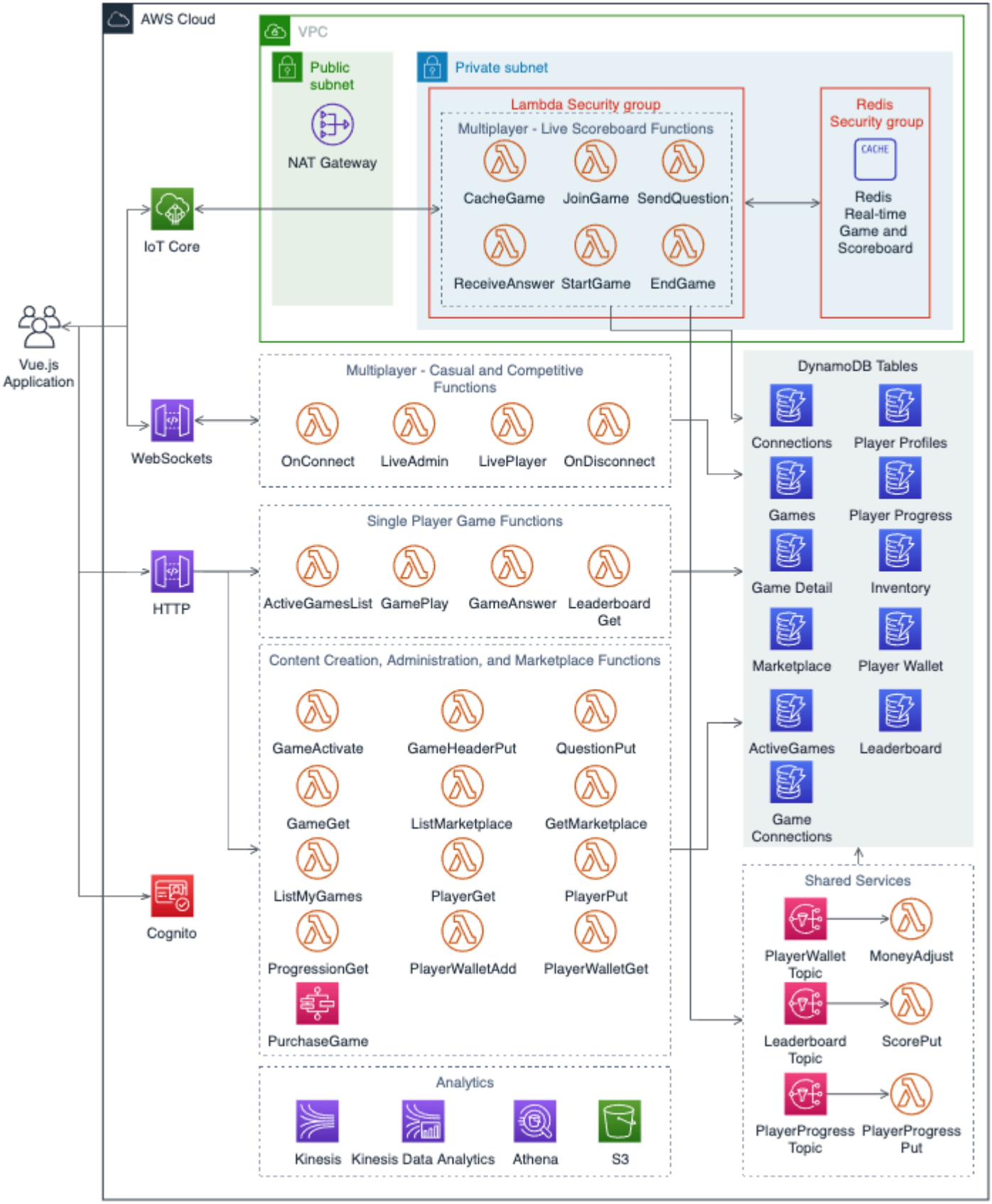
Ospitare un backend di gioco utilizzando contenitori

- I giocatori accedono al gioco utilizzando un software client di gioco, che può essere distribuito loro tramite un sistema di gioco, una vetrina digitale o un download diretto da una rete di distribuzione dei contenuti (CDN), come Amazon. CloudFront CDNsforniscono la memorizzazione nella cache nelle postazioni periferiche per accelerare le prestazioni degli utenti che scaricano contenuti. Ad esempio, CloudFront può essere utilizzato per distribuire il software client di gioco ai giocatori, nonché le risorse di gioco e altri contenuti.
- AWS Global Accelerator fornisce l'accelerazione del traffico e controlli personalizzabili per instradare il traffico dai client di gioco dei giocatori ai sistemi di bilanciamento del carico, nonché per instradare il traffico tra le regioni per scopi multiregionali e di failover. I nomi di dominio personalizzati per il backend di gioco REST APIs sono configurati in Amazon Route 53 per indirizzare il traffico verso gli endpoint Global Accelerator. Non mostrati nel diagramma,AWS Shield Advanced possono fornire un'ulteriore mitigazione DDo S per l'acceleratore e il backend di gioco.
- NAT Gateway e Application Load Balancer vengono distribuiti su sottoreti pubbliche in ciascuna delle zone di disponibilità utilizzate dal backend del gioco per fornire un'elevata disponibilità con la regione.AWS WAF viene distribuito su Application Load Balancer per fornire il filtraggio del traffico web di livello 7.
- Il backend del gioco è ospitato come una raccolta di singoli microservizi basati su container distribuiti in un cluster Amazon EKS in sottoreti private distribuite tra zone di disponibilità per garantire la resilienza. La scalabilità automatica regola dinamicamente la capacità dei servizi e dei nodi del cluster in base all'utilizzo delle risorse, che in genere è correlato alla domanda dei giocatori. Mentre Cluster Autoscaler regola automaticamente il numero di nodi nel cluster, Horizontal Pod Autoscaler ridimensiona automaticamente i pod distribuiti nel cluster.
- I dati di gioco e dei giocatori vengono archiviati in database e cache di backend che vengono distribuiti in sottoreti private nelle zone di disponibilità con replica tra nodi primari e di replica. Amazon Aurora è una scelta popolare per casi d'uso come profili dei giocatori, diritti e acquisti in-game, che possono avere requisiti di interrogazione più complessi e possono trarre vantaggio dalle funzionalità di modellazione dei dati relazionali di MySQL e PostgreSQL. Amazon ElastiCache è utile per creare classifere ad alte prestazioni, pub/sub messaggistica e per memorizzare nella cache i dati a cui si accede di frequente per ridurre la latenza e il carico sui database. Amazon DynamoDB è un data store NoSQL completamente gestito, ideale per modelli di accesso imprevedibili e la capacità di scalare fino a un throughput praticamente illimitato per casi d'uso come dati sullo stato di giocatori e partite, dati di sessione, inventario e archivi di articoli o casi d'uso in cui si desidera un database globale con un sovraccarico minimo.
- I flussi di lavoro di elaborazione asincrona devono essere utilizzati per eseguire attività che possono essere eseguite in background, come l'aggiornamento delle classiifiche o l'invio di

richieste di amicizia. Configura il backend di gioco per inserire questo tipo di lavoro nelle code di Amazon SQS e adattarlo alla crescita del gioco, oppure valuta la possibilità di utilizzare gli argomenti di Amazon SNS per distribuire il lavoro tra molte code di applicazioni consumer per l'elaborazione parallela. Usa le funzioni AWS Lambda per eseguire l'elaborazione in base agli eventi per ridurre i costi dell'infrastruttura di elaborazione e il sovraccarico di gestione. Per flussi di lavoro di lunga durata o che richiedono il coordinamento delle attività con più passaggi, prendi in considerazione l'idea di orchestrare l'intero flusso di lavoro utilizzando.AWS Step Functions Amazon EventBridge può essere utilizzato per avviare funzioni per rispondere a eventi di AWS servizio e applicazioni personalizzate.

Architettura di backend di gioco basata su server

Molti sviluppatori di giochi non vogliono gestire l'infrastruttura e preferiscono invece creare i propri giochi utilizzando tecnologie che consentano loro di concentrarsi sul software. In questo scenario è consigliata un'architettura serverless perché consente di creare e rilasciare funzionalità più rapidamente e con meno costi operativi. Le architetture serverless sono progettate utilizzando servizi cloud in grado di scalare dinamicamente in base alla richiesta senza la necessità di configurare, gestire e scalare i server. La seguente architettura di riferimento illustra come creare un gioco utilizzando un'architettura serverless.


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

Architettura di riferimento per il backend di gioco basata su server

Questa architettura di riferimento illustra un gioco a quiz basato sul web che offre funzionalità per giocatore singolo e multiplayer.

- Autenticazione del giocatore: i giocatori si autenticano utilizzando Amazon Cognito, che fornisce un'autenticazione sicura con una directory utente per la gestione delle identità dei giocatori.
- Logica di gioco come funzioni serverless: le funzionalità di gioco e la logica aziendale di backend vengono eseguite come AWS Lambda funzioni avviate in risposta a eventi, il che consente di ridurre i costi perché si pagano solo quando la funzione è in esecuzione. Lambda ti offre la flessibilità di scrivere ogni funzionalità di gioco come microservizio separato utilizzando un linguaggio di programmazione a tua scelta. Ad esempio, potresti scegliere di sviluppare funzioni.NET Lambda se hai esperienza nell'uso di C# per creare giochi Unity, oppure potresti scegliere di sviluppare le funzioni Lambda Node.js se desideri programmare un frontend e un backend per un gioco basato sul Web entrambi. JavaScript
- NoSQL Data Store per i dati di gioco e dei giocatori: utilizza DynamoDB per archiviare i dati dei giocatori e dei giochi, poiché è progettato appositamente per archiviare grandi quantità di dati dai microservizi. Come illustrato in questa architettura, è consigliabile utilizzare archivi dati separati per le esigenze di archiviazione dei dati di ciascuna funzionalità di gioco, il che semplifica il monitoraggio e la gestione delle funzionalità in modo indipendente. Ciò crea anche dei limiti di separazione se la proprietà delle funzionalità o dei servizi cambia all'interno del team. In questa architettura di riferimento, le tabelle DynamoDB vengono utilizzate per archiviare dati come lo stato della connessione, i dettagli del gioco, i progressi dei giocatori e le informazioni sulla classifica.
- Gameplay per giocatore singolo: le funzionalità per giocatore singolo consentono ai giocatori di eseguire azioni come selezionare e giocare una partita e visualizzare la classifica. Queste funzionalità sono implementate come servizi di RESTful backend ospitati con un'API HTTP Amazon API Gateway che richiama la funzione Lambda appropriata per ottenere e impostare dati nelle tabelle DynamoDB. Al termine del gioco, il backend invia anche notifiche agli argomenti di Amazon SNS che avviano le funzioni Lambda in modo asincrono per memorizzare i progressi e le statistiche del giocatore.
- Gameplay multigiocatore: le funzionalità di gioco multigiocatore richiedono che i giocatori siano in grado di interagire con il gioco per point-to-point comunicare, oltre a trasmettere e ricevere aggiornamenti da altri giocatori connessi. Un' WebSockets implementazione è adatta per la point-to-point comunicazione in un gioco leggero, come i quiz. I giocatori possono stabilire una WebSockets connessione ad Amazon API Gateway WebSockets, che gestisce la connessione e richiama le funzioni Lambda solo quando ci sono messaggi da inviare o ricevere per un giocatore.

Nei casi d'uso in cui è necessaria la one-to-many comunicazione tra giocatori, AWS IoT Core fornisce supporto per la messaggistica WebSockets tramite MQTT, che consente ai clienti di iscriversi agli argomenti e agire in base ai messaggi ricevuti. In questa architettura, gli WebSockets over MQTT vengono utilizzati per supportare casi d'uso come la trasmissione di aggiornamenti di gioco in tempo reale e per porre domande ai giocatori connessi. In alternativa AWS IoT, puoi scegliere Redis Pub/Sub per la consegna dei messaggi o Redis Streams se hai bisogno della conservazione dei messaggi.

- Usa le funzioni Lambda abilitate per VPC per accedere alle risorse nelle tue sottoreti private: configura le funzioni Lambda abilitate per VPC per accedere alle risorse nelle sottoreti private del tuo VPC, come ElastiCache Amazon, che viene utilizzato per ridurre i tempi di query per set di dati a bassa latenza come le classife live.

Per ulteriori informazioni, consulta la Guida per l'hosting di backend di gioco personalizzato su. AWS

Sviluppo di giochi su cloud (CGD)

Lo sviluppo di giochi su cloud (CGD) si riferisce all'infrastruttura e agli strumenti necessari per il ciclo di vita di sviluppo del gioco per creare, testare e sviluppare un gioco. Lo sviluppo del gioco è collaborativo tra gli utenti e i requisiti dell'infrastruttura cambiano spesso durante le fasi di sviluppo.

Molti sviluppatori di giochi stanno adottando team di sviluppo remoti e distribuiti a livello globale, il che richiede una tecnologia che supporti questo tipo di sviluppo. Gli sviluppatori di giochi possono ospitare tutti o parte di questi ambienti AWS e sfruttare la disponibilità globale di Regioni AWS per avvicinare le risorse agli utenti e gestire i loro ambienti di sviluppo in modo più conveniente scalando l'elaborazione e lo storage in base alle esigenze.

Gli ambienti possono variare a seconda delle esigenze degli sviluppatori di giochi, ma in genere includono postazioni di lavoro per sviluppatori per artisti, designer, ingegneri, tester di controllo qualità, appaltatori e altro personale per svolgere il proprio lavoro. Questi ambienti includono in genere anche una build farm composta da repository di codice sorgente per consentire agli utenti di registrare le modifiche e l' CI/CD infrastruttura per la creazione, l'imballaggio e il test degli artefatti sviluppati.

Queste architetture di produzione di giochi hanno le seguenti caratteristiche:

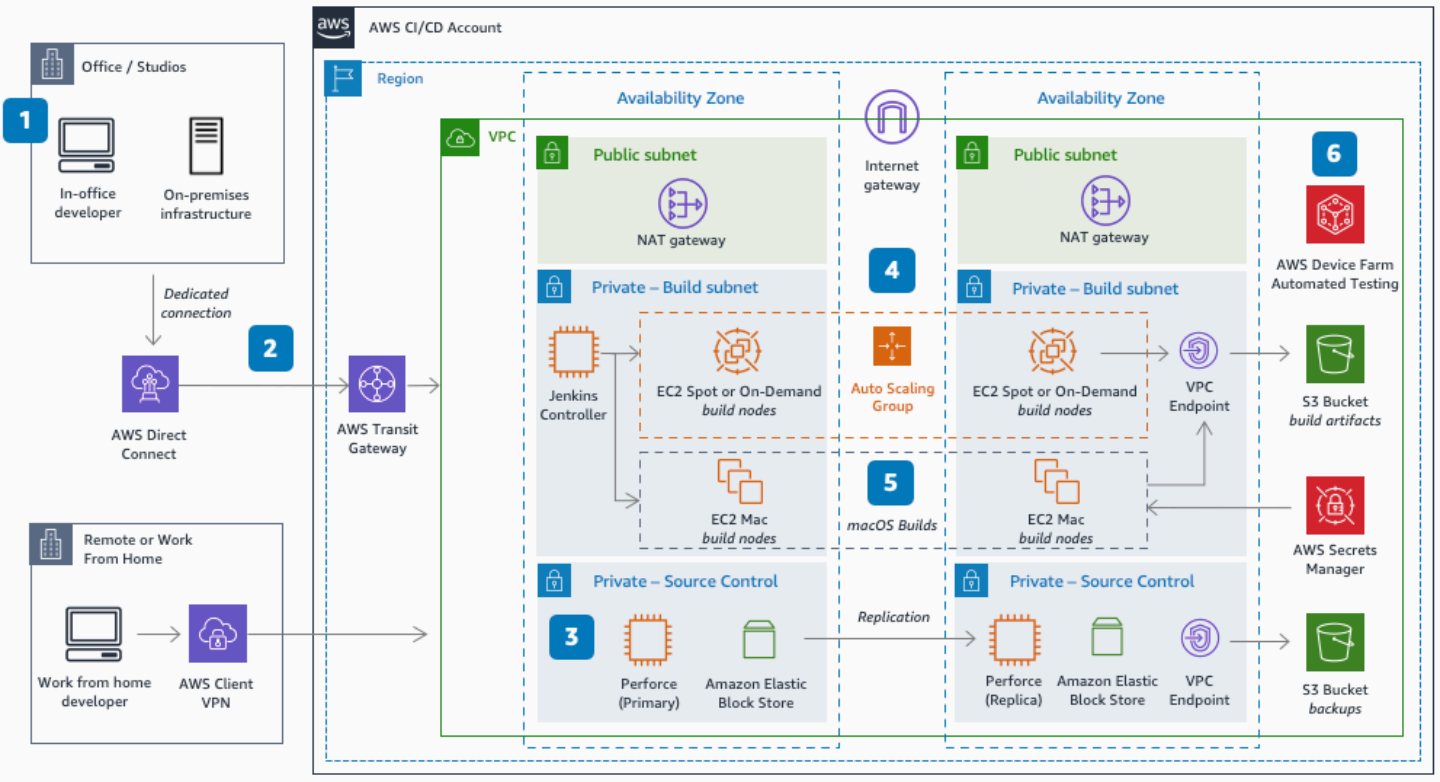
- Gli utenti dovrebbero essere in grado di accedere a una workstation virtuale tramite un browser Web o un client desktop locale, come Amazon DCV, che fornisce loro una sessione di streaming

a bassa latenza per accedere allo stesso software e agli stessi strumenti a cui avrebbero accesso se stessero lavorando su una macchina in un ufficio o in uno studio di sviluppo. Queste workstation virtuali, in genere un server basato su cloud, dovrebbero consentire a un utente di collaborare e lavorare sui propri progetti interamente in un ambiente cloud tramite LAN o WAN. Quando gli utenti non utilizzano attivamente le macchine, è necessario eseguire il backup del loro lavoro su un cloud storage durevole, ad esempio un repository di controllo del codice sorgente o un file system come Amazon Elastic File System (EFS) e Amazon FSx, e la loro macchina deve essere spenta per ridurre i costi.

- I repository di controllo del codice sorgente, come Perforce, devono essere progettati con un'elevata disponibilità utilizzando la replica tra zone di disponibilità o tra ambienti locali con backup archiviati in cloud storage come Amazon S3. Ad esempio, un server Perforce basato su cloud dovrebbe includere un server di commit primario ospitato in una zona di disponibilità con replica su un server di standby ospitato in un'altra zona di disponibilità nella stessa regione.
- Le risorse della build farm per lo sviluppo di giochi devono essere progettate con scalabilità automatica in modo che le risorse di elaborazione vengano fornite quando necessario, e le istanze EC2 Spot devono essere utilizzate per ridurre i costi derivanti dalla scalabilità orizzontale del numero di server necessari per le build.

Sviluppo di giochi su cloud: CI/CD

CI/CD l'infrastruttura è importante nello sviluppo di giochi indipendentemente dalle dimensioni del team per migliorare i tempi di iterazione, aumentare l'affidabilità, l'implementazione efficiente e controllare meglio il processo di sviluppo e rilascio per offrire ai giocatori un'esperienza di gioco di alta qualità. Una CI/CD pipeline di sviluppo di giochi è in genere composta da server e storage per il controllo del codice sorgente ad alta disponibilità, risorse di elaborazione per eseguire le build e software per eseguire test automatici, oltre alla corretta connettività di rete dei computer di sviluppo. La seguente architettura di riferimento dimostra come scaricare le build dei giochi da ambienti di sviluppo di giochi remoti o locali a quelle per aiutare gli sviluppatori a migrare o creare Cloud AWS nuove build farm.



Scarica le build dei giochi sul cloud

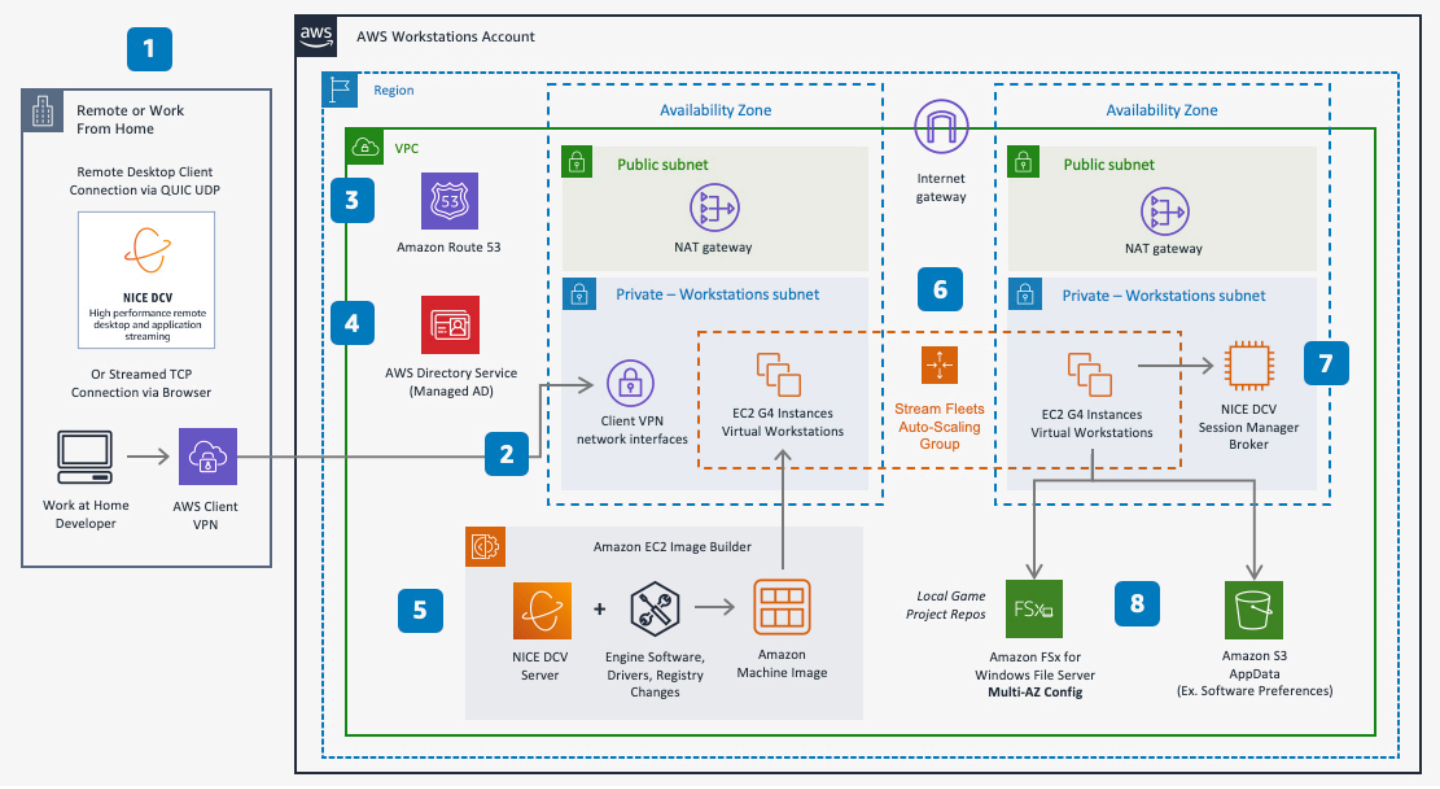
1. AWS Direct Connect fornisce una connessione privata, dedicata a bassa latenza AWS per gli sviluppatori in ufficio. Gli sviluppatori remoti utilizzano tecnologie zero-trust come Accesso verificato da AWS o reti private virtuali (VPN) come AWS Client VPN.
2. AWS Transit Gateway semplifica la gestione della rete per la connettività tra VPCs e da reti locali.
3. Perforce gestisce il controllo del codice sorgente e della versione (CI) supportato dallo storage Amazon EBS per dati persistenti e ad accesso rapido. Perforce Helix Core è disponibile in.Marketplace AWS
4. I commit avviano una build (CD) in Jenkins quando gli sviluppatori inviano modifiche a Perforce collegato a una filiale. Perforce avvia POST, un payload JSON su Jenkins. Il controller Jenkins richiama i comandi CLI headless del motore per eseguire e parallelizzare il processo di compilazione su nodi Docker temporanei (come Amazon Spot Instances o Amazon On-Demand Instances). EC2 EC2 Gli sviluppatori possono aumentare la disponibilità utilizzando due controller Jenkins, uno in ciascuna zona di disponibilità, con un sistema di bilanciamento del carico. Per alcuni motori di gioco, gli sviluppatori potrebbero aver bisogno di un'infrastruttura di licenza aggiuntiva configurata in sottoreti aggiuntive per vendere licenze per il contesto di compilazione ogni volta che viene eseguita una build simultanea.

5. La parte Xcode delle build iOS viene scaricata su istanze Amazon EC2 Mac per firmare, creare ed esportare il file.IPA, suddividendo il processo e riducendo i tempi di compilazione.Gestione dei segreti AWS contiene profili di provisioning, chiavi private e certificati.
6. Gli artefatti di build vengono consegnati ad Amazon S3, che invia notifiche di successo o fallimento. AWS Device Farm consente test automatici per dispositivi mobili.

Sviluppo di giochi su cloud: workstation

Lo sviluppo di giochi spesso implica team distribuiti che lavorano in remoto da varie località, richiedendo l'accesso a un'infrastruttura condivisa e la capacità di supportare uno sviluppo collaborativo e geograficamente dislocato. Questa tendenza verso un processo di sviluppo di giochi più distribuito, che include l'uso di work-for-hire studi e il lavoro a distanza, richiede l'implementazione di tecnologie e flussi di lavoro solidi per facilitare la produttività, la condivisione di competenze e pratiche di sviluppo agili.

La seguente architettura di riferimento dimostra come utilizzarla AWS per ospitare workstation remote di sviluppo di giochi utilizzando il protocollo Amazon DCV.



Trasmetti in streaming lo sviluppo di giochi da qualsiasi luogo con Amazon DCV

1. Amazon DCV è un protocollo di streaming che supporta lo streaming 4K a 60 FPS. Gli sviluppatori che utilizzano un browser si connettono tramite connessioni TCP, mentre i client desktop possono utilizzare QUIC UDP sulla porta 8443 per migliorare le prestazioni.
2. Gli sviluppatori utilizzano AWS Client VPN per una connessione sicura alle interfacce di rete nelle sottoreti delle workstation con traduzione degli indirizzi di rete di origine (SNAT).
3. Amazon Route 53 fornisce DNS privato per le risorse nel VPC e inoltro DNS in entrata e in uscita.
4. Directory Service fornisce Microsoft Active Directory gestito per consentire l'archiviazione locale dei progetti di gioco mappata ai singoli utenti.
5. Le workstation vengono create utilizzando un'Amazon Machine Image (AMI) creata con Image Builder. Le immagini includono Amazon DCV Server, software per sviluppatori, modifiche al registro e driver come i driver di gioco NVIDIA o i driver per periferiche. Marketplace AWS include quelli comunemente AMIs utilizzati per le workstation.
6. Le flotte di workstation utilizzano tipi di EC2 istanze grafiche di Amazon che forniscono GPUs e sono ridimensionate utilizzando gruppi di Auto Scaling. EC2
7. Amazon DCV Session Manager Broker consente la gestione di sessioni Amazon DCV.
8. L'archiviazione locale dei file dei progetti è ospitata in Amazon FSx for Windows File Server. Gli sviluppatori si impegnano a utilizzare una pipeline CI/CD separata passando dallo storage locale al controllo del codice sorgente.

Game Analytics Pipeline

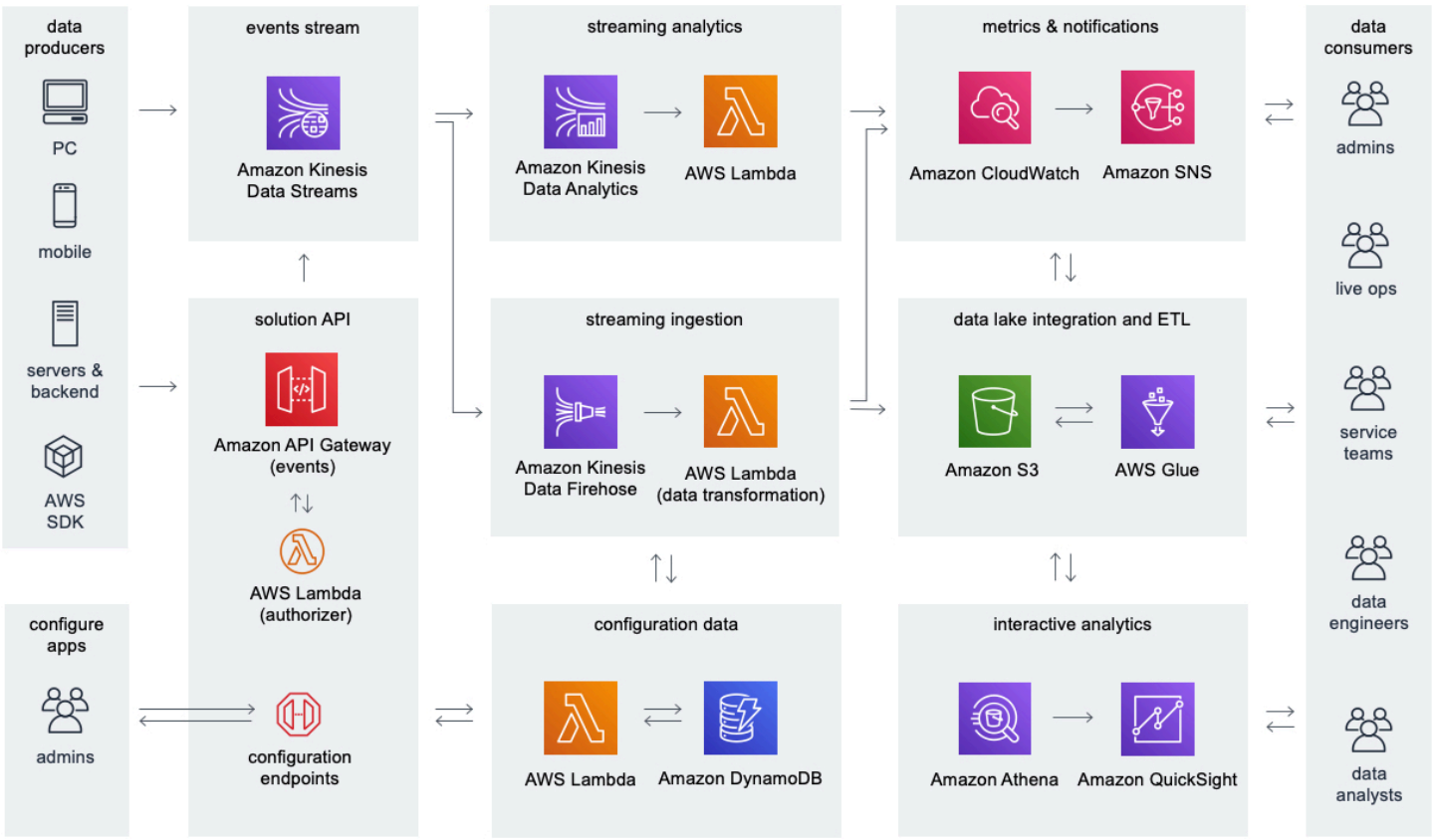
Gli sviluppatori di giochi sono sempre più alla ricerca di modi per comprendere meglio il comportamento dei giocatori in modo da poter migliorare l'esperienza di gioco per fidelizzare e far crescere la propria base di giocatori. L'analisi dei giochi rappresenta l'infrastruttura tecnica e i processi necessari per comprendere e analizzare i dati generati dal gioco e dai servizi correlati. Ciò richiede in genere l'uso di un'architettura di pipeline di analisi in grado di supportare questo end-to-end processo, come l'implementazione della soluzione Game Analytics Pipeline.

Le architetture di analisi dei giochi hanno le seguenti caratteristiche:

- Le fonti di dati inviano i dati in un formato comune come JSON e in genere includono server di gioco e servizi di backend per giochi, oltre a client di gioco, inclusi PC, dispositivi mobili e console di gioco.
- Una pipeline di analisi dei giochi automatizza l'intero flusso di lavoro di acquisizione e archiviazione dei dati grezzi e della loro elaborazione in formati di output utilizzabili in modo che possano essere

analizzati in modo efficiente ed economico dai consumatori di dati, come gli utenti finali e le applicazioni di analisi.

- Le pipeline di analisi dei giochi forniscono supporto per l'acquisizione e l'elaborazione di elevati volumi di dati in tempo reale su larga scala man mano che il gioco cresce.
- Fornisci supporto sia per i casi d'uso in tempo reale che per la creazione di report in batch. Ad esempio, le dashboard e gli avvisi in tempo reale vengono in genere utilizzati dai team operativi dal vivo per monitorare l'infrastruttura di gioco e il comportamento dei giocatori per rilevare problemi. I team di analisti dei dati in genere si affidano a report in batch e in base alle necessità per comprendere le tendenze nel tempo.



Pipeline di analisi dei giochi senza server per la telemetria del gioco

I dati di gioco vengono acquisiti da client di gioco, server di gioco e altre applicazioni. I dati di streaming vengono inseriti in Amazon S3 per l'integrazione dei data lake e l'analisi interattiva. L'analisi in streaming elabora eventi in tempo reale e genera metriche. I consumatori di dati analizzano i dati delle metriche in Amazon CloudWatch e gli eventi non elaborati in Amazon S3.

- API della soluzione e dati di configurazione: utilizza Amazon API Gateway per fornire un'API REST per amministrare la pipeline di analisi dei giochi e archiviare i dati di configurazione in Amazon DynamoDB utilizzando le funzioni Lambda. Puoi creare un portale interno sulla base di questa API o un'interfaccia a riga di comando personalizzata per l'amministrazione. Un'API REST fornisce anche l'autenticazione del server per importare dati di gioco da fonti di dati e inoltrare i dati di telemetria ad Amazon Kinesis Data Streams per l'elaborazione in tempo reale e l'inserimento nello storage.
- Flusso di eventi: Amazon Kinesis Data Streams acquisisce i dati di streaming dal gioco e consente l'elaborazione dei dati in tempo reale da parte di Amazon Data Firehose e Amazon Managed Service per Apache Flink.
- Analisi dello streaming: Managed Service for Apache Flink analizza i dati degli eventi di streaming da Kinesis Data Streams e può generare metriche e avvisi personalizzati che vengono pubblicati utilizzando le funzioni Lambda. CloudWatch
- Metriche e notifiche: usa Amazon CloudWatch per monitorare i parametri, i log e gli allarmi della tua soluzione. Usa Amazon SNS per inviare notifiche ai tecnici in servizio e ad altri consumatori di dati.
- Inserimento di streaming: utilizza Firehose per utilizzare i dati in streaming da Kinesis Data Streams e distribuirli al data lake in Amazon S3 per lo storage, la trasformazione e l'integrazione a lungo termine con altri dati.
- Integrazione con data lake ed ETL: utilizzala AWS Glue per i flussi di lavoro di elaborazione ETL e per organizzare i metadati in AWS AWS Glue Data Catalog, che fornisce la base per un data lake per l'integrazione con strumenti di analisi flessibili.
- Analisi interattive: gli utenti finali possono utilizzare Amazon Athena per eseguire query interattive ad hoc sui set di dati archiviati in Amazon S3 e Quick Suite può essere utilizzata per creare dashboard.

Fai riferimento alla Game Analytics Pipeline per un'implementazione di riferimento automatizzata di una pipeline di analisi che può essere distribuita nel tuo account utilizzando. AWS CloudFormation

Definizioni

Il AWS Well-Architected Framework si basa su sei pilastri: eccellenza operativa, sicurezza, affidabilità, efficienza delle prestazioni, ottimizzazione dei costi e sostenibilità. AWS fornisce diversi componenti principali che ti consentono di progettare state-of-the-art architetture per i tuoi carichi di lavoro di gioco. In questa sezione, presenteremo una panoramica delle definizioni chiave.

Ai fini di questo paper, un'architettura di gioco comprende l'infrastruttura tecnica di backend necessaria per creare e gestire un gioco. Alcuni giochi potrebbero non avere funzionalità social, multiplayer o altre funzionalità online e potrebbero non richiedere l'uso di alcuni aspetti dell'infrastruttura tecnica di backend descritti in questo paper. Per una discussione dettagliata dei diversi tipi di carichi di lavoro che vengono spesso utilizzati per supportare un'architettura di gioco, consulta Scenari.

L' Cloud AWS infrastruttura è costruita attorno a regioni e zone di disponibilità.

- Una regione è un'ubicazione fisica nel mondo in cui sono presenti più zone di disponibilità.
- Le zone di disponibilità sono costituite da uno o più data center discreti, ciascuno con alimentazione, rete e connettività ridondanti, ospitati in strutture separate.

A seconda delle caratteristiche del gioco, potresti voler implementare determinati componenti della tua architettura di gioco in più regioni per motivi come migliorare le prestazioni dei giocatori o offrire esperienze personalizzate ai giocatori a seconda della loro posizione.

Esistono molti tipi diversi di giochi e l'infrastruttura tecnica di backend richiesta per supportare un gioco varia a seconda del tipo di gioco in fase di sviluppo. Ad esempio, i tipi di giochi più diffusi possono includere sparatutto in prima persona (FPS), giochi di ruolo (RPG), giochi online multigiocatore di massa (MMOG), battle royale (BR), giochi sportivi, giochi di puzzle e altro ancora. Esistono anche diverse modalità di interazione di gioco che influenzano l'architettura del gioco, come il gioco a turni e il gioco simultaneo, con caratteristiche prestazionali diverse.

I giochi sono stati sviluppati per essere giocati su uno o più sistemi di gioco, tra cui desktop, web, dispositivi mobili, console e nuove modalità di interazione come realtà aumentata (AR), realtà virtuale (VR) e soluzioni di streaming di giochi. I giochi in genere supportano il gameplay tra sistemi diversi, il che significa che i giocatori possono salvare la progressione di gioco e riprendere il gioco su altri sistemi, nonché avviare sessioni di gioco con giocatori su altri sistemi.

La monetizzazione dei videogiochi consente agli editori di giochi di generare entrate utilizzando diverse strategie come pubblicità, acquisti di giochi digitali e al dettaglio, acquisti in-game di contenuti scaricabili (DLC) noti come microtransazioni e tramite gli abbonamenti a pagamento richiesti per giocare. Alcuni degli indicatori chiave di performance () più comuni nel settore dei giochi includono: KPIs

- Utenti attivi giornalieri (DAU)
- Utenti attivi mensili (MAU)
- Utenti simultanei (CCU)
- Durata della sessione
- Costo per installazione (CPI)
- Valore della durata del giocatore (LTV)
- Ricavo medio per utente (ARPU)

Sistema di gioco

I videogiochi sono stati sviluppati per essere giocati su un sistema di gioco che fornisce controlli di input del client, grafica, software client (noto come client di gioco) e hardware e, in alcuni casi, funzionalità esclusive del sistema per supportare il gioco.

I sistemi di gioco sono generalmente suddivisi nelle seguenti categorie:

- Console: sistemi di intrattenimento progettati appositamente per giocare, inclusi esempi popolari come Sony, PlayStation Microsoft Xbox e Nintendo Switch. Le console offrono la possibilità di giocare installando contenuti di gioco fisici o distribuiti digitalmente sull'hardware della console prodotto dal fornitore del sistema di gioco. In questa definizione, una console può essere portatile o fissa e destinata all'uso in uno scenario di intrattenimento domestico.
- Personal computer (PC): giochi giocati utilizzando un software per computer installato su un computer client che può essere personalizzato dal giocatore. Per questo motivo, i giochi per PC sono popolari tra i giocatori grazie alla flessibilità e al controllo che offrono.
- Web: giochi progettati per essere giocati utilizzando un browser Web e che di solito offrono il vantaggio di consentire a un giocatore di accedere al gioco indipendentemente dal sistema operativo.

- Dispositivi mobili: giochi sviluppati per essere giocati su un telefono cellulare, di solito un sistema operativo per smartphone. I giochi per dispositivi mobili vengono generalmente scaricati da un app store digitale e installati sul telefono.

Oltre ai sistemi citati in precedenza, esistono anche sistemi nascenti che sono ancora relativamente nuovi e in crescita e hanno una quota di mercato molto inferiore rispetto ai sistemi più diffusi. Esempi di sistemi di gioco di questa categoria includono AR, VR e lo streaming di giochi, a volte denominato cloud gaming.

Lo streaming di giochi prevede il rendering del gameplay nel cloud e lo streaming su un thin client, in genere un browser. Lo streaming di gioco consente a un giocatore di giocare a un gioco interamente ospitato in remoto, in genere nel cloud da un fornitore di servizi di streaming di giochi. Nello streaming di gioco, il giocatore si connette a un gioco basato sul cloud tramite un browser o un thin client fornito dal fornitore di servizi di cloud gaming (sistema di gioco).

Server di gioco

I server di gioco rappresentano uno degli aspetti più importanti dell'infrastruttura di elaborazione per il tuo gioco. I server di gioco, a volte denominati server di gioco dedicati, vengono utilizzati per lo sviluppo di un gioco multiplayer o quando è richiesta l'elaborazione autorevole degli eventi di gioco da parte del server. Il server di gioco è il fulcro dell'architettura di gioco e funge da luogo in cui viene eseguita la logica di base, che include la gestione del giocatore e dello stato del gioco, nonché la gestione delle interazioni tra i client di gioco connessi e il server di gioco. Il server di gioco è in genere uno degli aspetti più sensibili alle prestazioni di un'architettura di gioco, perché è responsabile dell'elaborazione degli input dal client di gioco di un giocatore e della loro corretta distribuzione agli altri giocatori connessi in tempo reale. Un server di gioco con prestazioni scadenti influisce sulle prestazioni complessive dell'esperienza di gioco. Pertanto, dovresti ottimizzare le prestazioni del server di gioco e fornire una capacità sufficiente, specialmente all'avvio del gioco o nei periodi di picco di gioco.

Ai fini del presente documento, per server di gioco o istanza di server di gioco si intende il computer, ad esempio una macchina virtuale, che ospita uno o più processi del server di gioco. Un processo del server di gioco rappresenta una singola istanza della build del server di gioco che ospita una sessione di gioco, ovvero un'istanza del gioco in esecuzione a cui i giocatori possono connettersi tramite una sessione di gioco. Per questo motivo, spesso ci riferiamo al processo del server di gioco o alla sessione di gioco in modo intercambiabile, a causa della relazione implicita uno a uno tra una sessione di gioco e il processo del server di gioco che la ospita. Inoltre AWS, sono disponibili diverse

opzioni di calcolo per ospitare server di gioco, che forniscono l'accesso a capacità scalabile basata sul cloud attraverso un provisioning elastico delle risorse.

Amazon EC2 fornisce server virtuali basati sul cloud, noti come istanze, con supporto per più versioni di Linux e Windows. Puoi creare istanze e gestirle direttamente come un altro server o macchina virtuale. In genere, su un'istanza vengono distribuiti più processi di server di gioco per migliorare l'efficienza e ridurre i costi. Amazon EC2 è un'ottima scelta per i server di gioco se desideri il massimo controllo sull'infrastruttura di elaborazione.

Amazon GameLift fornisce una soluzione completamente gestita per l'hosting di server di gioco dedicati nel cloud e funzionalità aggiuntive come il matchmaking con GameLift FlexMatch. GameLiftfornisce un livello di astrazione su Amazon EC2 per semplificare la gestione dei server di gioco ed è disponibile nella maggior parte dei casi, in Regioni AWS modo da poter ospitare server di gioco vicino ai giocatori per ridurre la latenza, ottenere un'elevata disponibilità e ridurre significativamente i costi utilizzando le istanze Spot. Sebbene GameLift possa essere integrato nei backend di gioco esistenti, è particolarmente utile per gli sviluppatori di giochi che non vogliono sviluppare le proprie soluzioni di gestione dei server di gioco e matchmaking e preferiscono una soluzione gestita AWS e scalabile man mano che il gioco cresce.

Amazon Elastic Container Service (Amazon ECS) è un servizio di orchestrazione di container completamente gestito che puoi utilizzare per eseguire contenitori basati su Docker. Puoi anche utilizzare Amazon Elastic Kubernetes Service (Amazon EKS) per eseguire container basati su Docker creati utilizzando Kubernetes. L'uso di tecnologie container come quelle fornite da Amazon ECS e Amazon EKS può aiutarti a migliorare l'utilizzo dell'elaborazione raggruppando in modo efficiente molti processi del server di gioco o altre istanze di applicazioni di gioco in un'istanza. EC2

L'uso dei container può anche migliorare la produttività degli sviluppatori ospitando applicazioni che utilizzano lo stesso runtime operativo dell'immagine Docker utilizzato dagli sviluppatori sulle loro macchine locali durante lo sviluppo. Puoi ridurre ulteriormente il sovraccarico operativo utilizzando AWS Fargate, che è una soluzione di elaborazione serverless per l'esecuzione di container ed è compatibile sia con Amazon EKS che con Amazon ECS. Fargate è la soluzione ideale per i casi d'uso in cui si desidera eseguire server di gioco in container senza la responsabilità di gestire le istanze sottostanti su cui vengono eseguiti i contenitori.

Puoi utilizzarlo AWS Outposts per eseguire AWS infrastrutture e servizi in un data center o in una struttura locale, il che può consentire l'esecuzione dei giochi in ambienti locali e AWS utilizzare la stessa infrastruttura per supportare una strategia di adozione del cloud ibrido. AWS Le Local Zones fungono da estensioni Regioni AWS che consentono ai server di gioco e ad altri carichi di lavoro sensibili alla latenza di funzionare più a stretto contatto con i giocatori o i team di sviluppo. Inoltre,

per ridurre la latenza di rete globale per i server di gioco, puoi utilizzare AWS Global Accelerator per migliorare le prestazioni del traffico dei giocatori verso i tuoi server di gioco.

AWS Lambda è un servizio di elaborazione senza server che esegue codice senza fornire o gestire server, il che lo rende utile per i casi d'uso asincroni dei server di gioco, come i giochi a turni o quelli che richiedono requisiti di elaborazione leggeri, una piccola base di codice e in cui le funzionalità di gioco possono essere progettate utilizzando un'architettura di microservizi stateless. È importante tenere presente che le funzioni Lambda vengono eseguite in base agli eventi e per richiesta, anziché eseguire un processo del server di gioco a esecuzione prolungata. Lambda fornisce l'astrazione più rapida delle opzioni illustrate in questo paper perché l'applicazione sottostante è immediatamente disponibile per gli sviluppatori tra cui scegliere per ospitare il proprio codice.

Quando scegli il tuo approccio per l'hosting di server di gioco, prendi in considerazione vari requisiti, tra cui sovraccarico operativo, codebase legacy, requisiti prestazionali e scalabilità. EC2le istanze e i contenitori sono ottime opzioni per le basi di codice legacy, in quanto richiedono la minima modifica per il passaggio al cloud e puoi utilizzare EC2 le istanze per dedicare le risorse di un'istanza di calcolo, mentre i container possono semplificare la gestione e l'elevato utilizzo. Le funzioni serverless offrono il massimo livello di astrazione, che puoi utilizzare per definire il codice che viene eseguito solo in risposta agli eventi, il che può ridurre i costi.

Client di gioco

Il client di gioco rappresenta il dispositivo software e hardware che il giocatore utilizza per giocare. Il client di gioco fornisce il software per tradurre gli input del giocatore in messaggi che vengono inviati a un server per l'elaborazione, ed è responsabile della gestione delle risposte in arrivo dal server e della trasmissione al giocatore di output, come la grafica. Nei giochi multigiocatore in rete in tempo reale, il client di gioco di solito mantiene una connessione di rete persistente a un server di gioco per tutta la durata di una sessione di gioco per ridurre la latenza di rete e ridurre al minimo i tempi di elaborazione. Tuttavia, il client di gioco può anche interagire tramite REST con un server di gioco o servizi di backend.

Messaggistica

In genere ci sono tre categorie principali di messaggi nei giochi:

- Messaggi di coinvolgimento dei giocatori rivolti a un utente o a una coorte di utenti specifici, come inviti a giochi o notifiche push

- Messaggi di gruppo tra giocatori, ad esempio chat all'interno del gioco
- ervice-to-serviceMessaggi S, come i messaggi JSON utilizzati per integrare due o più applicazioni

Una strategia comune per l'invio e la ricezione di questi tipi di messaggi consiste nell'utilizzare modelli di architettura di elaborazione asincrona e pubblicatore-abbonato. AWS offre diversi servizi che possono aiutarti a implementare la messaggistica nel tuo gioco.

- Amazon Simple Notification Service (SNS): servizio gestito per la distribuzione di messaggi tra editori e abbonati utilizzando un modello di architettura. pub/sub Gli editori inviano messaggi utilizzando un'API ad Amazon SNS, che consegna i messaggi in modo asincrono alle applicazioni sottoscritte e può inviare notifiche push direttamente a client o desktop mobili con supporto per alcuni dei servizi di notifica push più utilizzati. Amazon SNS può essere utilizzato per le notifiche push ai client e per i casi d'uso della service-to-service messaggistica.
- Amazon Simple Queue Service (SQS): un servizio di coda completamente gestito che semplifica l'integrazione dei server di gioco e del gioco indipendentemente dal linguaggio di programmazione utilizzato in ciascuno di essi. Molte attività di gioco possono essere disaccoppiate e gestite in background, come l'aggiornamento di una classifica o dei valori del tempo di gioco in un database. Questo approccio è un modo efficace per separare varie parti del gioco e scalare in modo indipendente le funzionalità rivolte ai giocatori dall'elaborazione in backend.
- Amazon Managed Streaming for Apache Kafka (MSK): un servizio completamente gestito che semplifica la creazione di streaming di dati e applicazioni per produttori o consumatori utilizzando Apache Kafka, una popolare soluzione open source. Kafka viene in genere utilizzato per l'acquisizione e l'elaborazione di dati di streaming in tempo reale e può essere utilizzato per la messaggistica. service-to-service
- Amazon ElastiCache (Redis OSS): fornisce un archivio dati in memoria completamente gestito che include il supporto per la popolare pub/sub funzionalità di Redis, comunemente utilizzata per lo sviluppo di applicazioni di chat room e messaggistica ad alte prestazioni. service-to-service Redis supporta anche tipi di dati complessi come elenchi e set, in modo che gli sviluppatori possano utilizzare Redis per accodamenti ad alte prestazioni.
- Amazon Pinpoint: fornisce messaggi di coinvolgimento degli utenti tramite e-mail, SMS, messaggi vocali e notifiche push. Ad esempio, Amazon Pinpoint può essere utilizzato per inviare messaggi di coinvolgimento degli utenti ai giocatori per invitarli a tornare al gioco e può essere utilizzato per casi d'uso transazionali come il supporto di token di autenticazione a più fattori, conferme d'ordine ed e-mail di reimpostazione della password.

Operazioni di gioco dal vivo (Live Ops)

Le operazioni di gioco dal vivo (Live Ops) sono uno stile di gestione e operazioni di gioco che considera un gioco come un servizio dal vivo e offre continuamente nuove funzionalità, aggiornamenti, promozioni, eventi di gioco e miglioramenti al gioco lanciato per migliorare l'esperienza della comunità di giocatori.

Tradizionalmente, i giochi venivano forniti come prodotti anziché servizi e spesso nuovi contenuti e funzionalità venivano incorporati nelle versioni o nei sequel successivi anziché nel prodotto lanciato. Con un approccio Live Ops alla gestione dei giochi, un team addetto alle operazioni di gioco può lanciare un gioco e mantenere una community di giocatori coinvolta attraverso sperimentazioni, promozioni, eventi di gioco e innovazioni per intrattenere i giocatori.

Sebbene questo approccio abbia il vantaggio di sbloccare nuove strategie di coinvolgimento dei giocatori e di generare flussi di entrate ricorrenti, richiede una maggiore esperienza operativa. Ad esempio, per implementare una strategia Live Ops di successo, uno sviluppatore potrebbe aver bisogno di integrarsi con i servizi cloud o gestire la propria infrastruttura tecnica di backend. Hanno anche bisogno di un modo efficace per identificare e rispondere ai problemi che sorgono nel gioco o all'interno della comunità di giocatori che possono influire negativamente sull'esperienza del giocatore.

Eccellenza operativa

Il pilastro dell'eccellenza operativa si concentra sulle migliori pratiche per l'implementazione e il funzionamento di giochi basati su cloud su larga scala. È importante concentrarsi sull'eccellenza operativa per mantenere un'esperienza positiva per i giocatori e implementare misure preventive per prepararsi e riprendersi da problemi che influiscono sulla loro esperienza.

Aree di interesse

- Principi di progettazione
- Operazioni in tempo reale
- Struttura dell'account
- Implementazioni di giochi
- Monitoraggio della salute
- Test di caricamento
- Ottimizzazione nel tempo
- Resources

Principi di progettazione

Oltre ai principi di progettazione del white paper Well-Architected Framework, i seguenti principi di progettazione possono aiutarti a raggiungere l'eccellenza operativa nella creazione e nella gestione dei giochi:

- Definisci obiettivi misurabili e raggiungibili per i team addetti alle operazioni di gioco e adattali se necessario: a causa della natura dei giochi basata sui successi, è difficile determinare in anticipo quanti giocatori giocheranno al tuo gioco al momento del lancio o quali aspettative avranno i giocatori per le tue operazioni di gioco in corso. È importante fissare obiettivi ambiziosi ma raggiungibili con le parti interessate e progettare un approccio che possa essere ampliato se il gioco supera le proiezioni e ridimensionato mentre i team di sviluppo del gioco ottimizzano l'esperienza del giocatore. Preparatevi adeguatamente e testate in anticipo per soddisfare questi requisiti e allineare gli stakeholder tecnici e aziendali agli obiettivi prefissati per la gestione del gioco. Una volta definiti gli obiettivi, i team di gioco possono raggiungere un equilibrio adeguato tra costi e prestazioni durante la pianificazione, la progettazione, il provisioning, il test, l'implementazione e la gestione dell'infrastruttura di backend del gioco.

- Utilizza i runbook operativi per pianificare la scalabilità delle attività legate al lancio di giochi e agli eventi speciali: i team addetti alle operazioni di gioco devono coordinarsi con gli stakeholder aziendali per modellare le proiezioni relative al previsto picco di partecipanti agli eventi ed eseguire una pianificazione proattiva per pre-scalare in anticipo la capacità dell'infrastruttura. A causa della natura fluttuante del traffico dei giocatori durante gli eventi, le attività di pianificazione e pre-scalabilità precedenti dovrebbero potenziare i sistemi di scalabilità automatizzati esistenti per aumentare le possibilità di successo durante un evento e verificare di disporre di risorse sufficienti per offrire un'esperienza positiva ai giocatori. Implementate procedure di ingegneria delle prestazioni per sviluppare una base delle risorse a vostra disposizione e una comprensione basata sui dati della capacità del sistema, che vi aiuterà a guidare le attività di pre-scalabilità e le configurazioni di scalabilità automatizzate. Sviluppa runbook operativi per garantire la coerenza del processo. Questa pianificazione anticipata e la reattività alla domanda dei giocatori sono particolarmente importanti per i giochi con servizio dal vivo, che devono mantenere prestazioni e infrastrutture affidabili per supportare una base di giocatori attiva e impegnata per un periodo di tempo prolungato.
- Stabilisci un modello operativo per ricevere, esaminare e rispondere alle richieste di assistenza dei giocatori: dopo il lancio, monitora le segnalazioni di reclami e problemi relativi al gioco. Implementa sistemi appropriati per interagire con i giocatori in modo sicuro ed efficace per risolvere adeguatamente i problemi dei giocatori, ad esempio forum di community, social media, e-mail, sistemi di biglietteria, call center o soluzioni automatizzate di chat bot. Ciò è particolarmente importante per i giochi con servizio live, che richiedono una comunicazione continua con la base di giocatori, la reattività al feedback dei giocatori per soddisfare le esigenze in evoluzione e il mantenimento di una comunità impegnata per un lungo periodo di vita.

Operazioni in tempo reale

GAMEOPS01: Come definisci la strategia delle operazioni live (Live Ops) del tuo gioco?

Sviluppa una strategia operativa dal vivo (Live Ops) per il tuo gioco basata su obiettivi e metriche prestazionali definiti, in consultazione con gli stakeholder aziendali.

Best practice

- GAMEOPS01-BP01 Usa gli obiettivi di gioco e le metriche delle prestazioni aziendali per sviluppare la tua strategia operativa dal vivo

GAMEOPS01-BP01 Usa gli obiettivi di gioco e le metriche delle prestazioni aziendali per sviluppare la tua strategia operativa dal vivo

Consulta gli stakeholder aziendali, come i produttori di giochi e i partner editoriali, per determinare gli obiettivi e le metriche prestazionali di un gioco. Questo può aiutarti a sviluppare piani per la gestione del gioco, tra cui la definizione delle finestre di manutenzione, i piani di aggiornamento del software e dell'infrastruttura e gli obiettivi di affidabilità e ripristinabilità del sistema.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Queste metriche possono anche aiutarti a determinare in quale fase del ciclo di vita del gioco dovresti incorporare uno stream operativo dal vivo (Live Ops) per monitorare lo stato del gioco, raccogliere feedback diretti sul gioco e creare processi di rilascio semplificati e automatizzati. Ad esempio, un nuovo gioco potrebbe attendere il raggiungimento di una determinata scala, misurata in base al numero di giocatori attivi, alle entrate o a un'altra serie di parametri, prima di creare un team operativo live dedicato. Uno studio di sviluppo di giochi affermato potrebbe avere già esperienza nelle operazioni dal vivo, magari per altri giochi, quindi dovrebbe solo aggiungere il nuovo gioco.

Passaggi dell’implementazione

- Puoi definire gli obiettivi per la concorrenza dei giocatori (CCU) e gli utenti attivi giornalieri e mensili (DAU e MAU) che l'infrastruttura di gioco dovrebbe essere in grado di supportare efficacemente, i budget per l'infrastruttura, gli obiettivi finanziari e altri obiettivi prestazionali, come la frequenza di rilascio di contenuti e funzionalità per aumentare il coinvolgimento dei giocatori. Questi obiettivi e metriche contribuiscono alle decisioni relative alla progettazione del gioco, alla gestione delle release, all'osservabilità e al supporto necessari per operazioni efficienti.
- Il tuo gioco potrebbe avere l'obiettivo di rilasciare nuovi aggiornamenti dei contenuti almeno una volta al mese senza tempi di inattività durante il rilascio. Queste informazioni ti aiutano a definire la strategia di distribuzione delle release e a coordinare la pianificazione della manutenzione richiesta, che potrebbe richiedere tempi di inattività in altri momenti del mese, e contribuire al raggiungimento del tuo SLA di disponibilità.

Struttura dell'account

GAMEOPS02: Come ti strutturi Account AWS per ospitare i tuoi ambienti di gioco?

Implementa una strategia multi-account per isolare diversi ambienti di gioco e migliorare la sicurezza, l'efficienza operativa e la scalabilità. Utilizzalo AWS Organizations per gestire le gerarchie degli account, applicare barriere agli account e applicare politiche di tag e tagging sulle risorse distribuite.

Best practice

- GAMEOPS02-BP01 Adotta una strategia multi-account per isolare giochi e applicazioni diversi nei propri account
- GAMEOPS02-BP02 Organizza le risorse dell'infrastruttura utilizzando l'etichettatura delle risorse

GAMEOPS02-BP01 Adotta una strategia multi-account per isolare giochi e applicazioni diversi nei propri account

Progetta una struttura di account che guidi l'implementazione dell'infrastruttura per soddisfare le esigenze di sicurezza, isolamento e operative di ogni ambiente. L'isolamento dell'ambiente, mediante la limitazione dell'accesso ad esso e la possibilità di utilizzare solo AWS i servizi necessari, è essenziale, in quanto gli ambienti di produzione sono bloccati, mentre gli ambienti di sviluppo e test sono favorevoli alla sperimentazione. Si consiglia vivamente di isolare ulteriormente i principali sottosistemi in ogni ambiente e di ospitare e gestire autonomamente i servizi comuni utilizzati da più ambienti. Account AWS

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Adotta una strategia multi-account AWS isolando i diversi ambienti (come sviluppo, test, staging, produzione e servizi condivisi) tra singoli ambienti, in modo da ridurre la Account AWS portata degli incidenti. Prendi in considerazione la possibilità di AWS Organizations gestire centralmente la gerarchia per Account AWS semplificare ulteriormente le operazioni, nonché di definire e applicare in modo selettivo le politiche a livello di account e di unità organizzativa (livello di unità organizzativa). Progettando un'unità Account AWS organizzativa e una struttura appropriate in linea con le esigenze del flusso di lavoro di sviluppo e produzione, è possibile ottimizzare i costi e migliorare la scalabilità.

- Adotta una strategia multi-account: isola gli ambienti per ridurre il raggio degli incidenti e semplificare le operazioni.
- Utilizzo AWS Organizations: gestisci gli account in modo gerarchico, applica le politiche e abilita la governance centralizzata.
- Piano per la scalabilità: progetta strutture di account dettagliate e implementa misure di riduzione dei costi per la crescita futura.

Passaggi dell'implementazione

Un sistema di gioco distribuito in AWS dovrebbe utilizzare più account organizzati in modo logico per garantire un isolamento adeguato, che riduce la portata dei problemi e semplifica le operazioni man mano che l'infrastruttura di gioco si espande. Account AWS le infrastrutture di gioco che ospitano i giochi sono in genere raggruppate nei seguenti ambienti logici:

- Gli ambienti di sviluppo del gioco vengono utilizzati dagli sviluppatori per sviluppare il software e i sistemi per il gioco.
- Gli ambienti di test o di garanzia della qualità (QA) vengono utilizzati per eseguire test di integrazione, controllo qualità manuale e altri test automatici che devono essere eseguiti.
- Gli ambienti di staging o di preproduzione vengono utilizzati per ospitare il software completo, in modo da poter eseguire test di carico e fumo prima del lancio in produzione.
- Gli ambienti live o di produzione vengono utilizzati per ospitare il software e l'infrastruttura live e servire il traffico di produzione proveniente dai giocatori.
- Gli ambienti di servizi o strumenti condivisi forniscono l'accesso a sistemi, software e strumenti comuni utilizzati da molti team diversi. Ad esempio, un repository centrale di controllo del codice sorgente e una game build farm ospitati autonomamente potrebbero essere ospitati in un account di servizi condivisi.
- Gli ambienti di sicurezza vengono utilizzati per consolidare i log centralizzati e le tecnologie di sicurezza utilizzate dai team che si concentrano sulla sicurezza del cloud.

Per l'infrastruttura di gioco AWS attiva, si consiglia di creare account separati per ogni ambiente di gioco (sviluppo, test, allestimento e produzione), nonché account per la sicurezza, la registrazione e i servizi condivisi centralizzati.

In genere, gli studi di sviluppo di giochi più piccoli che gestiscono un numero limitato di risorse infrastrutturali, in genere poche centinaia di server o meno, possono crearne uno Account AWS

per ciascuno di questi ambienti (ad esempio, un account di produzione, un account di sviluppo e un account di staging). Tuttavia, man mano che l'infrastruttura di gioco o le dimensioni del team aumentano nel tempo, questo modello semplificato potrebbe non adattarsi bene.

Quando configuri questi ambienti, tieni presente che molti AWS servizi condividono le Service Quotas a livello di risorse e API per un intero account all'interno di una particolare regione. Questo deve essere considerato quando si determina come organizzare gli account in modo logico. Account AWS devono sostenere i costi solo per il consumo dei servizi in essi distribuiti. Pertanto, ciò consente di ridurre efficacemente la contesa di risorse e le quote di servizio, in particolare man mano che il gioco cresce e sempre più sviluppatori hanno bisogno di accedere per creare e gestire le risorse.

In base alla nostra esperienza di collaborazione con grandi studi di sviluppo di giochi che in genere gestiscono migliaia di server e centinaia di sviluppatori accedono alle risorse, ti consigliamo di progettare una struttura di account più dettagliata in cui le singole applicazioni che supportano il gioco abbiano i propri account di sviluppo, test, staging e produzione. Poiché è difficile e dispendioso in termini di tempo riprogettare la strategia AWS multi-account dopo aver lanciato il gioco a causa della complessità della pianificazione e della migrazione dei sistemi live, prendi in considerazione le tue future esigenze di scalabilità nel determinare la giusta struttura multi-account.

È possibile utilizzarla AWS Organizationsper impostare una gerarchia e un raggruppamento di Account AWS unità organizzative e definirle (OUs) per applicare loro politiche comuni a livello di unità organizzative tramite politiche di controllo del servizio (). SCPs AWS Organizations gestisce e governa centralmente l'ambiente man mano che crescete e scalate le risorse. Puoi creare nuovi account e allocare risorse in modo programmatico, raggruppare account per organizzare i flussi di lavoro, applicare politiche di governance agli account o ai gruppi e semplificare la fatturazione utilizzando un unico metodo di pagamento per i tuoi account. Inoltre, Organizations è integrato con altri servizi in modo da poter definire configurazioni centrali, meccanismi di sicurezza, requisiti di audit e condivisione delle risorse tra gli account dell'organizzazione.

AWS Control Toweroffre un modo semplice per configurare e gestire un ambiente sicuro con più account, chiamato landing zone. Control Tower crea la tua landing zone utilizzando AWS Organizations, offrendo una gestione e una governance continue degli account, nonché le migliori pratiche AWS di implementazione basate sull'esperienza di lavoro con migliaia di clienti durante il passaggio al cloud. AWS ConfigAWS Trusted Advisor, e AWS Security Hub CSPMsono servizi che forniscono una visione aggregata o centralizzata dell'igiene del tuo account.

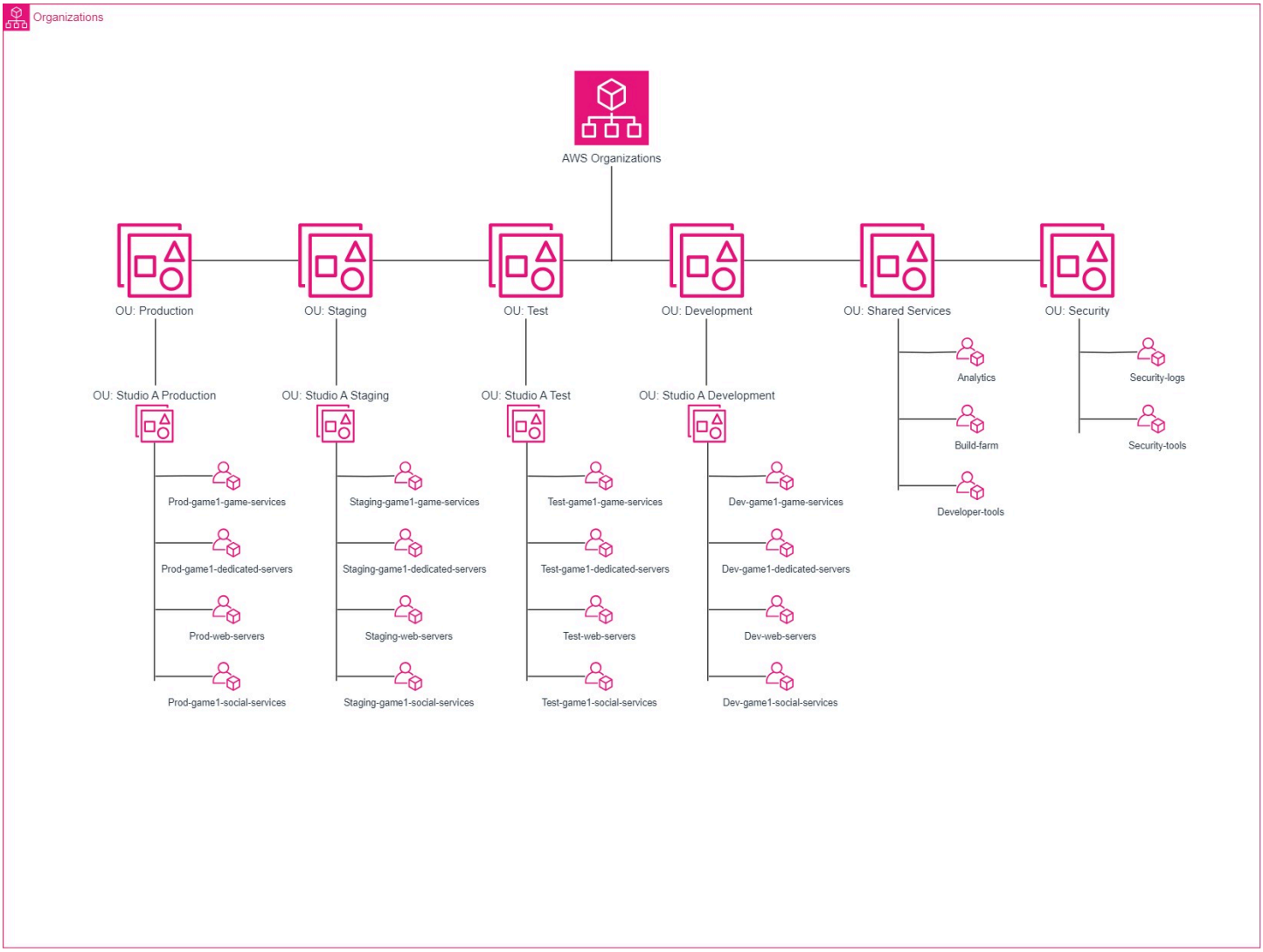
Questo isolamento ti aiuta a configurare autorizzazioni e barriere personalizzate o individuali per ogni ambiente di gioco. Gli account di produzione dovrebbero avere i guardrail, le restrizioni di accesso, il

monitoraggio e gli avvisi e gli strumenti di sicurezza necessari, mentre gli account non di produzione potrebbero non richiedere lo stesso livello di guardrail e autorizzazioni. Gli ambienti non di produzione possono essere automatizzati per disattivare le risorse fuori orario e ridurre i costi. La separazione degli account a questo livello di granularità semplifica il monitoraggio dei costi di infrastruttura per ciascuno degli ambienti che supportano un gioco.

Di seguito è riportato un esempio di struttura multi-account per una società di giochi che utilizza unità organizzative (OUs) per raggruppare Account AWS logicamente in ambienti AWS Organizations e studi separati. In questo esempio, OUs vengono utilizzati per raggruppare gli account in base al rispettivo ambiente e quindi in base allo studio che gestisce l'ambiente. Questo dimostra come è possibile creare una gerarchia annidata per consentire la distribuzione di applicazioni e giochi separati nei rispettivi account all'interno del rispettivo ambiente (illustrato come OUs), il che può essere utile se si sviluppano e gestiscono più giochi. Consulta la documentazione e i white paper forniti nella sezione risorse di questo pilastro per scoprire le strategie aggiuntive da prendere in considerazione per organizzare la tua strategia multi-account.

Sulla base della discussione precedente, il diagramma di esempio seguente presuppone uno studio di gioco (Organizzazione) con una pipeline di sviluppo composta da 4 fasi (sviluppo, test, messa in scena e produzione). Per un determinato gioco (game1), ogni ambiente (OU) dispone di servizi di gioco individuali, server Account AWS di gioco dedicati, servizi social e server web. Le risorse che vengono eseguite in ciascuno di essi Account AWS sono pertinenti ai rispettivi sottosistemi. In genere, ogni singolo gioco che utilizza questo tipo di pipeline di sviluppo replicherebbe questa struttura o una struttura simile. Account AWS

Oltre a questi ambienti incentrati sul gioco OUs, ci sono anche l'unità organizzativa dei servizi condivisi e l'unità organizzativa di sicurezza. Questi OUs dovrebbero riguardare l'intera organizzazione, non ogni singolo gioco. In questo modo i giochi consumerebbero i servizi condivisi per gli strumenti di sviluppo, i dati e l'analisi, come in questo esempio. Quindi, invia i registri delle applicazioni e del sistema alla Account AWS configurazione per i registri nell'unità organizzativa di sicurezza.



Esempio di struttura degli account per gli ambienti di gioco

GAMEOPS02-BP02 Organizza le risorse dell'infrastruttura utilizzando l'etichettatura delle risorse

Per gestire e tenere traccia in modo efficace delle risorse dell'infrastruttura AWS, utilizza l'etichettatura e il raggruppamento appropriati delle risorse per identificare il proprietario, il progetto, l'applicazione, il centro di costo e gli altri dati di ciascuna risorsa. Le risorse con tag possono essere raggruppate utilizzando gruppi di risorse, che facilitano il supporto operativo.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Definisci le politiche di etichettatura. Le strategie tipiche includono i tag delle risorse per identificare il proprietario della risorsa, come il nome del team o del singolo individuo, il nome del gioco, dell'applicazione o del progetto, il nome dello studio, l'ambiente (ad esempio sviluppo o produzione) e il ruolo della risorsa (come server di database, server web, server di gioco dedicato, server di app o server cache). Puoi aggiungere altri tag per soddisfare le esigenze aziendali e IT. AWS Configpuò anche applicare una politica di etichettatura al momento della creazione e dell'aggiornamento delle risorse. I tag e i gruppi di risorse sono disponibili nelle Console di gestione AWS operazioni AWS CLI, the e API.

Passaggi dell’implementazione

- Etichetta le risorse per identificarne il proprietario, il progetto, l'app, il centro di costo e altri dati pertinenti.
- Implementa politiche di etichettatura, tra cui tag per proprietario, progetto, studio, ambiente e ruolo della risorsa.
- AWS Config Utilizzalo per applicare le politiche di tagging e gestire i tag tramite Console di gestione AWS CLI e API.

Implementazioni di giochi

GAMEOPS03: Come gestisci le distribuzioni dei giochi?

Gestisci le implementazioni dei giochi convalidando accuratamente i componenti riutilizzati, eseguendo regolarmente la progettazione delle prestazioni e implementando test di carico regolari durante tutto il ciclo di vita dello sviluppo.

Best practice

- GAMEOPS03-BP01 Convalida e testa i sistemi e l'infrastruttura di gioco principali esistenti prima di riutilizzarli nel gioco
- GAMEOPS03-BP02 Esegui la progettazione delle prestazioni prima di ogni versione (o almeno per le versioni principali)
- GAMEOPS03-BP03 Test di caricamento precoce e frequente

- GAMEOPS03-BP04 Adotta una strategia di distribuzione che riduca al minimo l'impatto sui giocatori
- GAMEOPS03-BP05 È necessaria un'infrastruttura pre-scalabile per supportare i requisiti di picco

GAMEOPS03-BP01 Convalida e testa i sistemi e l'infrastruttura di gioco principali esistenti prima di riutilizzarli nel gioco

Le organizzazioni tendono a riutilizzare i componenti esistenti e il codice sorgente dei giochi precedenti per risparmiare tempi e costi di sviluppo. Questi componenti e codici legacy potrebbero non essere sottoposti a una revisione approfondita o essere sottoposti a test di integrazione dettagliati e basarsi invece sulle loro prestazioni passate.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Il riutilizzo aiuta a migliorare la produttività, ma può anche comportare il rischio di reintrodurre i precedenti problemi di prestazioni e stabilità in un nuovo progetto. Pertanto, quando si riutilizzano componenti esistenti e codice sorgente di giochi precedenti, è necessario implementare test affidabili.

Passaggi dell’implementazione

- Identifica codice e componenti riutilizzati: cataloga il codice sorgente, le librerie e i componenti riutilizzati dai giochi precedenti. Distingui chiaramente tra codice mantenuto attivamente e codice obsoleto
- Documenta il comportamento originale e i problemi noti: registra le caratteristiche prestazionali originali, i limiti funzionali e i bug o gli incidenti di produzione noti associati ai componenti riutilizzati.
- Esegui una revisione approfondita del codice: esegui una revisione tecnica dettagliata dei componenti riutilizzati, in particolare quelli che presentavano problemi in passato o che sono scarsamente documentati.
- Sostituisci o rifattorizza i componenti legacy ad alto rischio: dai priorità alla sostituzione o all'aggiornamento dei componenti legacy che presentano una storia di problemi o non sono più gestibili, anziché affidarti a soluzioni alternative in produzione.
- Esegui test di integrazione e compatibilità: convalida i componenti riutilizzati nel contesto dei sistemi del nuovo gioco. Verifica che interagiscano correttamente con nuovi moduli, strumenti e. APIs

GAMEOPS03-BP02 Esegui la progettazione delle prestazioni prima di ogni versione (o almeno per le versioni principali)

L'ingegneria delle prestazioni è il processo di monitoraggio di più metriche operative chiave di un'app per scoprire opportunità di ottimizzazione che possono migliorare ulteriormente le prestazioni dell'applicazione. Si tratta di un processo iterativo che inizia con il test, seguito dall'ottimizzazione del codice, delle sue dipendenze, dei processi associati, del sistema operativo host e dell'infrastruttura sottostante.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Per condurre un'analisi più approfondita delle prestazioni dell'app, integra nel codice dell'app uno strumento di monitoraggio delle prestazioni delle applicazioni (APM) o di debug in grado di isolare i problemi e ridurre i tempi di risoluzione dei problemi monitorandone il comportamento per rilevare eventuali anomalie nei flussi dell'app. Gli strumenti APM sono anche in grado di identificare metodi e operazioni esterne con prestazioni lente.

AWS X-Rayassiste gli sviluppatori nelle loro attività di ingegneria delle prestazioni, come l'identificazione dei punti deboli delle prestazioni e l'analisi e il debug degli errori di produzione. È possibile utilizzare X-Ray per comprendere le prestazioni dell'applicazione e dei relativi servizi sottostanti e identificare e risolvere la causa principale dei problemi e degli errori di prestazioni. Attraverso numerosi round di test di carico, in cui l'applicazione e la relativa infrastruttura vengono caricate gradualmente con il traffico sintetico dei giocatori, vengono identificati vari colli di bottiglia del sistema, errori delle app, eccezioni, problemi del sistema operativo e altri problemi che potrebbero non essere stati rilevati durante altri test di controllo qualità.

Per eventi critici come il lancio di giochi, il rilascio di contenuti, le promozioni e i principali eventi di gioco, usa AWS Countdown, che fornisce linee guida all'implementazione basate su playbook creati da esperti di giochi per verificare la prontezza operativa, mitigare i potenziali rischi e pianificare le esigenze di capacità.AWS Countdown offre anche un'opzione di supporto premium che offre supporto avanzato e opzioni come gli ingegneri per ottimizzare l'infrastruttura.

Passaggi dell’implementazione

- L'ingegneria delle prestazioni implica la valutazione e il monitoraggio delle principali metriche operative per verificare che il codice, i processi, il sistema operativo e l'infrastruttura

dell'applicazione funzionino come previsto. La revisione pre-produzione aiuta anche a definire le prestazioni di base a diversi livelli di utilizzo simulato.

- Scopri e monitora metriche chiave come utilizzo, servizi, I/O, processi e così via utilizzando strumenti di sistema come sar, top, vmstat, sysstat, netstat e Performance Monitor.
- Tieni traccia delle prestazioni e del comportamento dell'applicazione utilizzando strumenti APM per isolare i problemi, identificare i colli di bottiglia ed eseguire il debug degli errori di produzione AWS X-Ray .
- Per eventi critici come il lancio di giochi, abbonati a AWS Countdown (IEM) per ricevere indicazioni architettoniche e operative, supporto operativo su richiesta, identificare i rischi e pianificare le mitigazioni.

GAMEOPS03-BP03 Test di caricamento precoce e frequente

Il test di carico è il processo di simulazione del traffico reale su un sistema per valutarne l'affidabilità e le prestazioni.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Il test di carico è un fattore chiave per lo sviluppo di una base prestazionale per le risorse e la comprensione della capacità del sistema, che può guidare le previsioni finanziarie, la progettazione dell'architettura, l'allocazione delle risorse, le configurazioni di scalabilità automatizzate e le attività di pre-scalabilità post-lancio. I vantaggi aggiuntivi includono:

- Infrastruttura ottimizzata: il provisioning delle risorse potrebbe essere eccessivo o insufficiente. La comprensione delle risorse necessarie comporterà una riduzione dei costi e una minore infrastruttura da gestire.
- Predisposizione alla scalabilità: alcuni meccanismi e funzionalità possono spingere gli utenti a iniziare rapidamente a giocare. Sapere quando e come scalare può fare la differenza tra soddisfare adeguatamente l'aumento della domanda e perdere giocatori. Utilizza i risultati dei test di carico per preparare i runbook con soglie di sistema, punti di avviso e punti di avviso critici a diversi livelli di scala.
- Codice di qualità superiore: problemi come l'eccessiva diafonia tra i servizi, le chiamate al database non in batch, gli algoritmi inefficienti, le perdite di memoria e i problemi di degrado del servizio sono a volte più semplici da identificare su larga scala.

- Convalida del comportamento: l'inserimento di diversi tipi di errori nei test può convalidare il comportamento previsto del sistema o scoprire problemi di gestione degli errori che devono essere corretti.

Idealmente, gli sviluppatori dovrebbero eseguire i test di carico in più punti durante il processo di sviluppo, poiché ognuno di essi può offrire vantaggi diversi: nella fase iniziale, guidano le decisioni sull'architettura e le attività di refactoring, mentre è più economico e semplice apportare modifiche. Alla fine di ogni sprint o iterazione, convalidano le prestazioni dell'applicazione con le caratteristiche e le funzionalità più recenti.

Prima dell'implementazione in produzione, i test di carico su larga scala che simulano i modelli di utilizzo previsti nel mondo reale confermano la capacità del sistema di gestire il carico di lavoro di produzione. Dopo l'implementazione, i test di carico periodici monitorano le prestazioni del sistema e identificano i cambiamenti o le strozzature che possono insorgere nel tempo.

Per simulare il traffico dei giocatori, sono necessari client o bot leggeri che emulino i flussi del client di gioco e che interagiscano con il backend del gioco per simulare il comportamento dei giocatori nel mondo reale. Questi dati vengono generalmente acquisiti attraverso i log di gioco e i dati generati da test di controllo qualità condotti da persone, nonché attraverso alfa o beta test reali su scala limitata in cui i giocatori reali sono invitati a giocare a una versione del gioco ad accesso anticipato.

È importante registrare il comportamento del sistema in un runbook operativo per facilitare la risoluzione di possibili guasti futuri e conservare i parametri prestazionali con cui confrontare i test di carico futuri. Si consiglia inoltre di far testare il gioco da personale addetto al controllo qualità durante il test di carico, in quanto potrebbe scoprire problemi che i bot non riescono a identificare e che le metriche non riflettono.

AWS Fault Injection Service è un servizio completamente gestito per l'esecuzione di esperimenti di iniezione dei guasti che semplificano il miglioramento delle prestazioni, dell'osservabilità e della resilienza di un'applicazione. Gli esperimenti di iniezione dei guasti vengono utilizzati nell'ingegneria del caos, ovvero la pratica di stressare un'applicazione in ambienti di test o di produzione creando eventi dirompenti, come un aumento improvviso del consumo di CPU o memoria, osservando la risposta del sistema e implementando miglioramenti. Gli esperimenti di iniezione dei guasti aiutano i team a creare le condizioni reali necessarie per scoprire i bug nascosti, monitorare i punti ciechi e le strozzature prestazionali difficili da individuare nei sistemi distribuiti.

Passaggi dell’implementazione

- Configura un ambiente di test di carico distribuito utilizzando Guidance for Kubernetes-Bases Game Load Testing.
- Personalizza e distribuisci Locust Control e Worker Pods all'interno del cluster EKS utilizzando i file di distribuzione forniti, abilitando una generazione di carico scalabile e gestibile.
- Registra il comportamento e le metriche del sistema durante i test di carico in un runbook operativo per facilitare la risoluzione dei problemi futuri e stabilire le linee di base delle prestazioni.
- Utilizza esperimenti di fault injection per simulare interruzioni nel mondo reale e scoprire problemi nascosti nelle prestazioni, nell'osservabilità e nella resilienza del sistema.

GAMEOPS03-BP04 Adotta una strategia di distribuzione che riduca al minimo l'impatto sui giocatori

Incorpora una strategia di implementazione per il software e l'infrastruttura di gioco che riduca al minimo i tempi di inattività che impediscono ai giocatori di giocare. Sebbene alcuni tipi di aggiornamenti possano richiedere l'installazione di nuovi aggiornamenti nel client di gioco, progetta il gioco in modo da ridurre al minimo o evitare i tempi di inattività durante le distribuzioni.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Uno dei passaggi più importanti da considerare quando si sviluppa una strategia di distribuzione del gioco è determinare come verrà gestita l'infrastruttura di gioco. Gestisci la tua infrastruttura di gioco utilizzando uno strumento Infrastructure as Code (IaC) come AWS CloudFormationTerraform di Hashicorp per ridurre gli errori umani durante la preparazione dell'ambiente. I modelli di infrastruttura possono essere implementati e testati in pipeline automatizzate, il che crea coerenza nella configurazione dei diversi ambienti di gioco.

Esistono diverse strategie di distribuzione che possono essere utilizzate per un gioco:

Sostituzione rotativa

L'obiettivo principale di una sostituzione a rotazione in caso di schieramento è effettuare il rilascio senza interrompere il gioco e senza influire sui giocatori. È importante che l'aggiornamento o le modifiche da eseguire siano compatibili con le versioni precedenti e funzionino in modo adiacente alle versioni precedenti del sistema.

In questa distribuzione, le istanze del server vengono sostituite in modo incrementale (sostituite o implementate) da istanze che eseguono la versione aggiornata. Questa sostituzione progressiva può essere eseguita in diversi modi. Ad esempio, per implementare aggiornamenti continui a una flotta di server di gioco dedicati, un approccio tipico prevede la creazione di un nuovo gruppo di EC2 istanze Auto Scaling contenente la nuova versione di build del server di gioco distribuita su di esse, e quindi il routing graduale dei giocatori verso sessioni di gioco ospitate su questo nuovo parco di server. Se è associato un aggiornamento del client di gioco richiesto come prerequisito per utilizzare la nuova build del server di gioco, devi includere un controllo di convalida per verificare che solo i giocatori su cui è installato questo nuovo aggiornamento del client di gioco vengano indirizzati a queste sessioni di gioco.

Le flotte di server (ad esempio, i gruppi di EC2 Auto Scaling) contenenti la vecchia versione di build del server di gioco vengono rimosse dal servizio solo dopo aver esaurito le sessioni attive dei giocatori in modo corretto, in genere impostando metriche personalizzate del server che consentono ai team addetti alle operazioni di gioco di automatizzare questo processo. In alternativa, per ridurre la quantità di infrastruttura e il tempo necessario per eseguire un'implementazione continua, è possibile adottare un approccio alternativo in cui le istanze di produzione esistenti vengono rimosse dal servizio, aggiornate con la nuova build del server di gioco e quindi reinserite nel parco macchine di produzione. Questo approccio riduce la quantità di infrastruttura richiesta, ma aumenta anche il rischio, poiché il numero di server di gioco live disponibili per i giocatori viene ridotto man mano che i server vengono sostituiti.

Questo modello può essere utilizzato anche per eseguire distribuzioni continue su servizi di backend come database, cache e server di applicazioni che non ospitano il gameplay. Purché questi servizi siano distribuiti in modo altamente disponibile con più istanze cluster, la complessità delle implementazioni su questi servizi dovrebbe essere inferiore a quella delle implementazioni su server di gioco dedicati.

Implementazione blu/verde

L'obiettivo principale di una blue/green distribuzione in un gioco è ridurre al minimo i tempi di inattività, garantendo al contempo un ripristino sicuro della distribuzione precedente in caso di problemi. È adatto per implementazioni in cui due versioni del backend di gioco sono compatibili e possono servire i giocatori contemporaneamente.

Nella strategia blue/green di distribuzione, vengono configurati due ambienti identici (blu e verde). La versione del gioco esistente è etichettata in blu, mentre la nuova versione del gioco, che rappresenta l'obiettivo di distribuzione, è etichettata in verde. Quando l'ambiente verde è pronto per la migrazione, è possibile configurare il livello di routing in modo da trasferire il traffico verso l'ambiente verde,

mantenendo al contempo disponibile il vecchio ambiente (blu) nel caso in cui sia necessario un failback. In questo scenario, gli aggiornamenti del routing potrebbero richiedere l'aggiornamento del servizio di matchmaking per configurarlo per iniziare a inviare sessioni di gioco alla nuova flotta o, nel caso dei servizi di backend di gioco, potrebbe riguardare l'aggiornamento dei record DNS in Amazon Route 53 per il tuo servizio o lo spostamento dei pesi del load balancer delle applicazioni per inviare traffico al nuovo gruppo target.

Uno degli svantaggi della strategia di blue/green implementazione è il costo intrinseco dell'ambiente di standby dovuto all'infrastruttura aggiuntiva richiesta durante l'esecuzione dell'installazione. Un'opzione per mitigare questo costo aggiuntivo dell'infrastruttura consiste nel prendere in considerazione l'adozione di una variante di blue/green implementazione in cui il nuovo software di gioco venga distribuito sugli stessi server già distribuiti in produzione. In questo scenario, è possibile avviare un nuovo processo server verde con il nuovo software accanto al processo server blu esistente, con il passaggio tra i processi del server anziché tra un'infrastruttura fisica separata. Questo approccio può anche velocizzare l'implementazione dei giochi su una grande quantità di infrastruttura, eliminando la necessità di attendere il lancio di nuovi server nel cloud. Per le best practice su questo approccio di implementazione, consulta Blue/Green Deployments on. AWS

Implementazione delle Canarie

L'implementazione di Canary è utile per gli sviluppatori di giochi, in quanto la strategia può essere applicata per rilasciare una versione alpha o beta anticipata di un gioco o una funzionalità di gioco come una nuova modalità di gioco, mappa o sfida a un gruppo ristretto o ristretto di giocatori in produzione. Tale schieramento si chiama canarino. La versione potrebbe includere funzionalità di tracciamento e segnalazione aggiuntive, quindi quando giocatori reali giocano a quel gioco o a quella funzionalità, i dati di telemetria di gioco vengono raccolti e analizzati per individuare eventuali anomalie e problemi.

Per quanto riguarda le nuove funzionalità, i giocatori non vengono costantemente informati in merito e la telemetria del gioco è la fonte principale utilizzata per determinare se i giocatori stanno riscontrando problemi e la versione deve essere annullata. Allo stesso tempo, se non vengono identificati problemi significativi, la funzionalità può essere ulteriormente estesa a più giocatori per ottenere dati aggiuntivi. Se i giocatori vengono avvisati, può essere chiesto loro di fornire un feedback regolare sulla loro esperienza. Tale attività di test sarebbe idealmente coordinata da un team operativo in diretta.

Come strategia, Canary Deployment può essere utilizzato anche per le versioni standard per rendere gradualmente disponibile una nuova funzionalità ai giocatori. Un potenziale vantaggio rispetto all' blue/green ambiente standard è che non è richiesto un secondo ambiente completo. La capacità del

nuovo ambiente ridimensionato determina il numero di giocatori che devono essere integrati nella nuova funzionalità. Prima di aggiungere altri giocatori, la capacità deve essere ridimensionata in modo appropriato. Anche se si prevede che questa blue/green tecnica personalizzata avrà un costo comparativamente inferiore rispetto alla tecnica blu/verde standard, si stima comunque che comporti costi che potrebbero essere superiori a quelli della tecnica di sostituzione a rotolamento dei canarini.

Utilizzate un solo canarino in un ambiente di produzione e concentratelo per raccogliere dati e feedback. Se vengono utilizzati più canarini, ciò complica la risoluzione e l'isolamento dei problemi di produzione e compromette la qualità dei set di dati e del feedback raccolti.

Una variante del canary si verifica quando uno o più esperimenti (generalmente test dell'interfaccia utente) vengono eseguiti tramite implementazioni mirate, in cui un set di server di backend di gioco serve una versione di una funzionalità e un altro set delle stesse dimensioni serve un'altra versione della stessa funzionalità. A tale scopo non viene creata alcuna infrastruttura aggiuntiva o speciale e solo le aree prescelte dei server di backend ricevono questi aggiornamenti. Il risultato degli esperimenti consiste nell'osservare come i giocatori reagiscono a ciascuna delle versioni della stessa funzionalità, determinare se esiste un consenso generale di gradimento o antipatia e osservare se ci sono problemi identificati con l'usabilità o la funzionalità. Tali esperimenti strategici sono anche chiamati A/B test e l'intero processo è chiamato test A/B. Al termine di questi esperimenti, vengono raccolti i dati di test necessari prima di tornare alla versione corrente del sistema di backend di gioco sui server utilizzati per i test.

Implementazioni tradizionali precedenti

Nello stile di implementazione tradizionale, durante una finestra di manutenzione programmata il gioco viene chiuso e i giocatori connessi vengono abbandonati o esauriti prima che le istanze del server all'interno del backend di gioco vengano aggiornate con le ultime build di codice. Questa distribuzione ha un impatto sui giocatori ogni volta che viene eseguita e i giocatori devono essere avvisati prima del programma. Di conseguenza, questo modello causa il maggior impatto sui giocatori e dovrebbe essere evitato quando possibile.

Dopo l'installazione dell'aggiornamento, è possibile testare il gioco prima di aprirlo ai giocatori, che aspetterebbero la riapertura del gioco. Ciò può causare un picco di traffico quando i giocatori cercano di accedere e giocare entro un breve periodo di tempo. Pertanto, se il gioco non è progettato per gestire tali picchi di traffico, puoi scegliere di consentire ai giocatori di rientrare gradualmente in gioco in batch.

In alternativa, puoi optare per un sovradimensionamento dell'infrastruttura per sostenere il picco di traffico iniziale e, una volta che il traffico di gioco si sarà stabilizzato, le risorse potranno essere

ridotte. Se necessario, esegui questo tipo di schieramento durante le ore non di punta, quando il numero di giocatori è minimo. La manutenzione programmata frequentemente, così come la manutenzione prolungata, comporta intrinsecamente un rischio di abbandono dei giocatori e una potenziale perdita di entrate. I giocatori si aspettano cambiamenti anche dopo una nuova versione e possono perdere la fiducia nel gioco una volta rientrati dopo un periodo di inattività.

Passaggi dell'implementazione

- Riduci al minimo i tempi di inattività: implementa strategie di implementazione che riducano i tempi di inattività e mantengano i giocatori impegnati nel gioco.
- Infrastructure as code (IaC): utilizza strumenti come AWS CloudFormation o Terraform per gestire l'infrastruttura di gioco e ridurre gli errori umani.
- Strategie di distribuzione: utilizza una o una combinazione di implementazioni a rotazione sostitutiva, blu/verde e canarino per fornire aggiornamenti fluidi e ridurre l'impatto sui giocatori.

GAMEOPS03-BP05 È necessaria un'infrastruttura pre-scalabile per supportare i requisiti di picco

Scalate l'infrastruttura in vista di eventi di gioco su larga scala per assicurarvi di poter gestire l'improvviso aumento della domanda da parte dei giocatori.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Oltre al lancio di nuovi giochi, le partite dal vivo in genere prevedono eventi in-game, promozioni, nuovi contenuti e uscite stagionali come esempi di modi per sostenere e migliorare il coinvolgimento dei giocatori. Tali attività registrano un elevato volume di traffico di giocatori per tutta la durata dell'evento o della promozione. L'azienda si aspetta di raggiungere o superare gli obiettivi prefissati per l'evento e l'infrastruttura di gioco deve sostenerla e supportarla durante tutto questo.

Prepara la tua infrastruttura in anticipo per essere in grado di supportare il carico di giocatori previsto durante eventi su larga scala. Per prepararsi, i team addetti alle operazioni di gioco devono coordinarsi con le parti interessate alle vendite e al marketing per stimare la domanda prevista che verrà generata in un evento imminente esaminando la concorrenza dei giocatori passati, le metriche di coinvolgimento e i dati di vendita. Se l'evento prevede il lancio di un nuovo gioco, i team addetti alle operazioni di gioco dovrebbero collaborare con queste parti interessate per identificare proiezioni

realistiche sulla scala prevista. Sebbene possa essere difficile prevedere il successo di un gioco, è importante che tutti comprendano quali sono le aspettative di successo, in modo che l'infrastruttura possa essere scalata e testata per supportare tali obiettivi.

Molti giochi scelgono di essere lanciati a tappe, iniziando con un soft launch, aprendo il gioco a un numero limitato di giocatori e poi ampliando in modo organico i giocatori in ogni fase, prima del lancio pubblico completo. Durante il periodo di soft launch, monitora, identifica, traccia e risolvi i problemi mentre perfezionate le vostre proiezioni per il lancio pubblico.

Per stimare correttamente i requisiti dell'infrastruttura, raccogli dati tramite test di carico e prestazioni eseguiti sui backend di gioco in esecuzione in un ambiente di produzione o simile alla produzione prima del lancio del gioco. È necessario eseguire più round di questi test per simulare diverse condizioni del gioco e verificare che il backend sia in grado di sopportare il carico nella maggior parte delle condizioni.

Per raggiungere questo obiettivo, gli sviluppatori possono scrivere bot di gioco che utilizzano vari flussi di lavoro del gioco ed emulano condizioni diverse. Questi test dovrebbero ispezionare i diversi livelli di sistema del backend di gioco in modo che ogni livello e componente venga testato e i dettagli vengano registrati. Usa i dati raccolti da questi test per preparare il piano per il lancio del gioco.

I punti di errore singoli (SPOF) devono essere identificati e rimossi ove possibile, rendendo l'applicazione altamente disponibile e tollerante ai guasti. Utilizzate i test di carico per effettuare l'identificazione SPOFs emulando i guasti a diversi livelli upstream e downstream e verificando il comportamento del gioco e degli altri componenti.

Oltre all'infrastruttura stimata necessaria per il lancio del gioco, l'evento di gioco o la preparazione della promozione, configura il sistema in modo da scalare automaticamente su richiesta. Definisci, configura e monitora le soglie di scalabilità degli eventi per consentire al backend di gioco di scalare per sostenere un elevato volume di traffico di giocatori. Per il traffico variabile, il pre-provisioning è la soluzione migliore perché potrebbe non esserci abbastanza tempo per la scalabilità orizzontale. Potrebbe essere necessario il ridimensionamento manuale durante il lancio iniziale del gioco, in modo da soddisfare una domanda superiore al previsto più rapidamente di quanto i sistemi automatizzati siano in grado di scalare le risorse.

Sì AWS, le organizzazioni dovrebbero richiedere Service Quotas più elevate per i servizi che utilizzano nel backend di gioco. I Service Quotas sono impostati per gli account per evitare che i clienti installino o scalino inavvertitamente un'infrastruttura più ampia del previsto. Quando un gioco in esecuzione su un account raggiunge il limite superiore della quota di servizio configurata in quella regione, il servizio limita le richieste oltre la quota assegnata e interrompe le disposizioni. Le manette

possono causare errori involontari o imprevisti e compromettere l'esperienza del giocatore. Monitora, monitora e rivedi regolarmente le soglie di quota di servizio per i servizi utilizzati dal gioco in fase di produzione per evitare limitazioni. Quando l'utilizzo supera una soglia di quota di servizio tollerabile, è possibile richiedere un aumento della quota inviando un Support Case dal Console Support Center, dopo aver effettuato l'accesso all'account interessato o utilizzando l'API Support.

Per eventi critici come lanci di giochi, rilasci di contenuti, promozioni ed eventi di gioco importanti, usa Countdown.AWS Countdown fornisce linee guida per l'implementazione basate su playbook creati dagli esperti di Games per garantire la prontezza operativa, mitigare i potenziali rischi e pianificare le esigenze di capacità.AWS Countdown offre anche un'opzione di supporto premium che offre supporto avanzato e opzioni come gli ingegneri per ottimizzare l'infrastruttura.

Se stai lanciando un gioco ospitato su Amazon GameLift, consulta le liste di controllo pre-lancio per prepararti.

Passaggi dell’implementazione

- Scalate anticipatamente l'infrastruttura: preparate in anticipo l'infrastruttura per eventi di gioco su larga scala per gestire gli aumenti improvvisi della domanda da parte dei giocatori.
- Stima della domanda: coordinati con le vendite e il marketing per stimare la domanda prevista utilizzando i dati passati dei giocatori e proiezioni realistiche.
- Test di carico e rimozione degli SPOF: esegui più cicli di test di carico per convalidare la capacità del backend, identificare singoli punti di errore e configurare correttamente la scalabilità automatica.

Monitoraggio della salute

GAMEOPS04: Come si monitora lo stato di salute del gioco?

Monitora lo stato del gioco implementando una strumentazione completa per rilevare e tenere traccia dei problemi che influiscono sui giocatori, tra cui la registrazione delle attività sul lato client, il monitoraggio del servizio di backend e la segnalazione degli errori. Utilizza una combinazione di AWS strumenti come Amazon CloudWatch e AWS X-Ray soluzioni di terze parti per identificare e risolvere rapidamente i problemi.

Best practice

- GAMESOPS04-BP01 Strumenta il gioco per rilevare e monitorare i problemi che influiscono sui giocatori

GAMESOPS04-BP01 Strumenta il gioco per rilevare e monitorare i problemi che influiscono sui giocatori

Oltre a rispondere ai social media e alle segnalazioni di problemi dei giocatori, offri al tuo gioco soluzioni di monitoraggio per rilevare e indagare sui problemi che influiscono sui giocatori.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Nessun test è in grado di identificare tutti i problemi di un gioco. I giochi vengono generalmente lanciati con problemi noti che dovrebbero essere risolti gradualmente con la prossima versione del gioco. I problemi noti e riproducibili sono semplici da risolvere. Per facilitare l'identificazione di tali problemi, i client di gioco devono implementare il monitoraggio delle attività dei giocatori, la registrazione delle app e la reportistica in vari punti strategici per aiutare il team di backend a identificare i problemi lato client. La possibilità di individuare tempestivamente tali problemi aiuta gli sviluppatori del gioco a risolverli e risolverli prima che si diffonda. I dati e i registri riportati dal codice di tracciamento non devono mai includere informazioni di identificazione personale (PII) e devono contenere solo metadati specifici del gioco che facilitano il debug.

Implementa una soluzione di osservabilità per rilevare e rispondere a problemi come arresti anomali o bug del gioco. Puoi usare Amazon CloudWatch Synthetics per creare canarini in grado di monitorare lo stato dei tuoi servizi di gioco di backend rivolti ai giocatori. Puoi utilizzare i tuoi servizi di backend AWS X-Rayper tracciare le richieste tra i servizi distribuiti e inviare log e metriche personalizzati ad Amazon. CloudWatch

Le soluzioni di terze parti, come Backtrace.io e Sentry, sono soluzioni popolari per la segnalazione degli errori nei giochi. Anche le soluzioni di monitoraggio delle prestazioni delle applicazioni (APM) di partner come New Relic, Splunk, Datadog e Honeycomb.io sono popolari.

Oltre ai canali di supporto ufficiali, il team operativo del gioco e i community manager dovrebbero inoltre monitorare vari social network e canali per verificare i feedback dei giocatori, i reclami e le segnalazioni di bug. Esamina e prova a riprodurre ogni reclamo specifico del gioco oppure invialo al team addetto al controllo qualità per la revisione. Se riproducibile, inoltra il problema agli sviluppatori del gioco per la risoluzione dei problemi e la soluzione prima che abbia ripercussioni sulla base di giocatori più ampia.

Passaggi dell'implementazione

- Implementa soluzioni di monitoraggio: utilizza gli strumenti di monitoraggio per rilevare i problemi che influiscono sui giocatori e rispondere rapidamente.
- Tieni traccia delle attività e dei registri dei giocatori: utilizza i client di gioco per registrare le attività dei giocatori e segnalare problemi e verificare che non siano incluse informazioni di identificazione personale (PII).
- Utilizza AWS strumenti e strumenti di terze parti: utilizza strumenti come CloudWatch X-Ray e soluzioni di terze parti per la segnalazione degli errori e il monitoraggio delle prestazioni e monitora i social media per il feedback dei giocatori e le segnalazioni di bug.

Test di caricamento

GAMEOPS05: Cosa dovresti considerare durante il test di carico di un gioco?

Durante il test di carico di un gioco, considera la fase di test, l'architettura di generazione del carico e il framework di test appropriati per valutare efficacemente le prestazioni e la scalabilità del sistema. Scegli la giusta combinazione di tempistica (sviluppo iniziale, sprint, pre-produzione o post-implementazione), infrastruttura (EC2EKS, Fargate o Lambda) e strumento di test (JMeterLocust, Grafana K6 o Gatling) per allinearti alle caratteristiche uniche e agli obiettivi di sviluppo del tuo gioco.

Best practice

- GAMEOPS05-BP01 Scegli lo stage, l'architettura e il framework di test di carico giusti per raggiungere i tuoi obiettivi

GAMEOPS05-BP01 Scegli lo stage, l'architettura e il framework di test di carico giusti per raggiungere i tuoi obiettivi

L'approccio al test di carico di un gioco può variare in modo significativo a seconda di molti fattori, tra cui la fase del processo di sviluppo in cui viene eseguito, l'architettura del sistema di generazione del carico stesso e la scelta del framework di test del carico. Il momento in cui verrà condotto, nelle fasi iniziali, durante gli sprint iterativi, prima dell'implementazione in produzione o dopo l'implementazione, determinerà gli obiettivi e il focus delle attività di test. I diversi modelli di infrastruttura di generazione del carico hanno i propri pro e contro e la scelta del framework di test

del carico influenza notevolmente le funzionalità, la facilità d'uso e le integrazioni disponibili per il processo di test. Allineando attentamente questi elementi, i team di sviluppo possono adattare l'approccio ai test di carico alle caratteristiche uniche del gioco, estrarre le informazioni più preziose sulle prestazioni e fornire un'esperienza fluida ai propri giocatori.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Test di carico in diverse fasi di sviluppo

L'esecuzione di test di carico esplorativi nelle prime fasi di sviluppo può convalidare l'architettura di sistema sottostante. Questo aiuta gli sviluppatori a prendere decisioni informate sull'infrastruttura del gioco, sulla progettazione del database e sulla topologia di rete prima che venga svolto un ampio lavoro di implementazione. I test di carico identificano i rischi e creano una base di riferimento per le prestazioni, riducendo potenzialmente al minimo la necessità di costose rilavorazioni e debiti tecnici più avanti nel ciclo di vita dello sviluppo. Possono anche favorire una comprensione condivisa dei requisiti prestazionali del gioco tra il team, portando a una migliore collaborazione e processo decisionale. In definitiva, i test di carico durante le fasi iniziali gettano una solida base per un gioco ad alte prestazioni, scalabile e resiliente, contribuendo a migliorare l'esperienza complessiva del giocatore.

Al termine di ogni sprint o iterazione, i test di carico possono valutare l'impatto sulle prestazioni delle nuove funzionalità, delle correzioni di bug e di altre modifiche introdotte nell'ultimo ciclo. Questo approccio mirato consente ai team di sviluppo di identificare rapidamente le regressioni o i peggioramenti delle prestazioni introdotti dagli ultimi aggiornamenti, consentendo loro di risolvere questi problemi prima che si propaghino ulteriormente nella pipeline e mantenendo un livello costante di qualità e prestazioni.

Prima di passare alla produzione, dei robusti test di carico aiutano i team a convalidare la capacità del sistema di gestire le condizioni di traffico e carico previste nel mondo reale. Possono individuare gli ostacoli alla scalabilità o i limiti di risorse all'interno dell'infrastruttura di produzione e offrire l'opportunità di ottimizzare le prestazioni del gioco, creando un'esperienza utente fluida e reattiva sin dal primo giorno. Le informazioni acquisite dai test di carico prima del lancio possono mitigare i rischi del giorno del lancio e contribuire alla pianificazione continua della capacità, che pone le basi per la sostenibilità e la scalabilità a lungo termine del gioco.

Il test di carico di un gioco già in produzione consente ai team di monitorare le prestazioni del gioco e identificare regressioni o peggioramenti delle prestazioni che possono verificarsi nel tempo. Ciò consente loro di risolvere i problemi in modo proattivo prima che influiscano sull'esperienza del

giocatore e influiscano negativamente sulla fidelizzazione degli utenti. Inoltre, i test di carico in produzione convalidano l'efficacia degli sforzi di ottimizzazione delle prestazioni o della scalabilità dell'infrastruttura implementati. Questo processo offre ai giocatori un'esperienza di gioco di alta qualità, reattiva e scalabile anche quando il gioco si evolve e matura.

Architetture che generano carico

La progettazione dell'architettura di generazione del carico per i test di carico di gioco può assumere varie forme, ognuna con una serie di vantaggi e considerazioni.

Al livello più elementare, le EC2 istanze Amazon autogestite possono essere fornite e configurate per fungere da generatori di carico. Con un approccio basato su nodi di controllo e nodi di lavoro, puoi configurare più istanze che generano carico, ognuna delle quali esegue il proprio script di test e gestita complessivamente da un'unica istanza di controllo. L'architettura può scalare e generare più carico senza aumentare la complessità creando nodi di lavoro aggiuntivi, ma questo approccio pratico richiede ai team di gestire il provisioning, la configurazione e la gestione dell'infrastruttura sottostante.

Per un approccio più scalabile e orchestrato, puoi utilizzare i cluster Amazon EKS Kubernetes per gestire e distribuire il carico di lavoro di test del carico su una flotta di agenti di carico basati su container. Le funzionalità di scalabilità automatica di Kubernetes possono essere utilizzate per gestire il ridimensionamento dei pod che generano carico, mentre i team stessi configurano e gestiscono le istanze sottostanti nel cluster che ospita i pod. EC2

In alternativa, la natura serverless di AWS Fargatepuò velocizzare e semplificare la configurazione dei test di carico eliminando la gestione dell'infrastruttura, pur garantendo la scalabilità e la flessibilità necessarie. Per le soluzioni ibride in cui esiste già un cluster Kubernetes locale che genera carichi, ma potrebbe essere necessaria una capacità aggiuntiva, EKS Anywhere è in grado di gestire entrambi i cluster come se fossero uno dal. Console di gestione AWS

È inoltre possibile utilizzare AWS Lambdale funzioni in base alle proprie esigenze e ai propri obiettivi. Le funzioni Lambda sono relativamente semplici da configurare e scalare senza la necessità di fornire e gestire risorse aggiuntive. Consentono inoltre la creazione di scenari di test più complessi e dinamici grazie alla profonda integrazione con altri servizi. AWS Tuttavia, le funzioni Lambda presentano limiti alle funzioni simultanee e al tempo di esecuzione (15 minuti), il che può limitare la scala e la durata dei test di carico che è possibile eseguire. Anche le latenze di avvio a freddo possono influire sulla precisione dei risultati e le limitazioni delle risorse di Lambda potrebbero non essere adatte a carichi di lavoro di test di carico molto impegnativi.

Gli studi che desiderano utilizzare una soluzione preconfigurata possono utilizzare Distributed Load Testing su. AWS Questa soluzione utilizza Amazon ECS AWS Fargate per distribuire container in

grado di eseguire simulazioni di decine di migliaia di utenti connessi. È possibile utilizzarlo per avviare rapidamente l'infrastruttura di test di carico in stile IAC utilizzando. AWS CloudFormation

Framework di test di carico

Non esistono due framework di test di carico creati allo stesso modo. Alcuni dispongono di interfacce grafiche intuitive per la creazione di test, mentre altri sono interamente basati sulla riga di comando. Uno strumento può essere flessibile e performante ma richiedere tempo e impegno per la configurazione e la gestione, mentre un altro potrebbe essere serverless ma limitato nei test che può creare ed eseguire. Alcuni godono di grandi community e di numerosi tutorial, pur non essendo collaudati sul campo, a differenza di altri che, pur essendo già ampiamente testati in fase di produzione, non dispongono del supporto o della documentazione della comunità. Scegli il framework che rappresenta il giusto equilibrio per te e il tuo team. Alcune delle opzioni più popolari sono:

- Apache JMeter: popolare framework di test di carico open source basato su Java grazie al suo robusto set di funzionalità e alla facilità d'uso. La sua capacità di simulare scenari utente complessi, l'ampia gamma di protocolli supportati, la reportistica completa e la comprovata esperienza lo rendono JMeter una scelta affidabile per i test di carico.
- Locust: framework di test di carico moderno e distribuito basato su un'architettura basata sugli eventi, che lo rende performante ed efficiente sotto il profilo delle risorse. I test sono scritti in Python, il che consente scenari di test flessibili che sfruttano migliaia di potenti librerie di terze parti, pur rimanendo intuitivi e semplici da leggere.
- Grafana K6: potente framework di test di carico che combina facilità d'uso con funzionalità avanzate. Il supporto per la generazione di carichi distribuiti, lo scripting flessibile e la perfetta integrazione con Grafana per la visualizzazione dei dati rendono Grafana K6 una scelta interessante.
- Gatling: framework di load testing open source noto per le sue prestazioni e scalabilità. Il suo linguaggio DSL (Domain-Specific Language) basato su Scala consente agli sviluppatori di creare script di test di carico concisi e gestibili, e le sue solide funzionalità di reporting e analisi forniscono informazioni dettagliate sul sistema in esame.

Passaggi dell’implementazione

- Fasi di test di carico: esegui test di carico in varie fasi di sviluppo (sviluppo iniziale, sprint, preproduzione e post-implementazione) per convalidare le prestazioni del sistema e identificare i problemi.

- Architetture di generazione del carico: scegli le architetture di generazione del carico appropriate (EC2EKS, Fargate o Lambda) in base alle esigenze di scalabilità, alle preferenze di gestione e ai requisiti di test specifici.
- Framework di test di carico: seleziona un framework di test di carico (come Locust JMeter, Grafana K6 o Gatling) che bilanci facilità d'uso, prestazioni, flessibilità e supporto della community per soddisfare le esigenze del tuo team.

Ottimizzazione nel tempo

GAMEOPS06: Come ottimizzi il gioco nel tempo?

Ottimizza il tuo gioco nel tempo monitorando le metriche chiave e i dati di telemetria per identificare le tendenze dei giocatori, le prestazioni del sistema e le aree di miglioramento. Aggiorna continuamente il design del gioco, l'infrastruttura e l'approccio ai test di carico sulla base di queste informazioni, adattandoti al contempo a nuove tecnologie e framework per offrire prestazioni ed esperienza di gioco ottimali man mano che il gioco si evolve.

Best practice

- GAMEOPS06-BP01 Monitora le metriche chiave del gioco per identificare le tendenze e i modelli dei giocatori e utilizza le informazioni per migliorare il gioco
- GAMEOPS06-BP02 Aggiorna e adatta l'approccio al test di carico man mano che il gioco cambia

GAMEOPS06-BP01 Monitora le metriche chiave del gioco per identificare le tendenze e i modelli dei giocatori e utilizza le informazioni per migliorare il gioco

Oltre all'utilizzo del sistema client di gioco, all'utilizzo delle app, alle eccezioni e ai dati sugli arresti anomali, acquisisci i dati di telemetria del gioco che vengono inviati a un sistema di backend di gioco. Questi dati devono rappresentare l'attività del giocatore in modo da poter capire come i giocatori interagiscono con le varie funzionalità del gioco.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

A seconda dell'implementazione, i client di gioco possono raccogliere dati di telemetria in luoghi o funzioni di gioco predefiniti in un mondo di gioco. I dati vengono inviati al servizio di acquisizione di backend per l'elaborazione. Se il servizio di backend non è raggiungibile, i client possono archiviare i dati localmente sul dispositivo locale fino a quando il servizio di backend non sarà nuovamente disponibile. I game designer utilizzano questi dati di telemetria per verificare come i giocatori stanno giocando e se ci sono anomalie nel gioco.

Ad esempio, i movimenti e le interazioni dei giocatori con gli elementi di una mappa possono essere estratti dai dati di telemetria e tracciati come mappa termica delle attività di gioco dei giocatori in un determinato intervallo di tempo. Questi dati aiutano i progettisti del gioco a identificare la necessità di bilanciare vari elementi del gioco, come la potenza di un'arma, la potenza di un personaggio in gioco o la complessità di una mappa. I dati di telemetria grezzi vengono generalmente archiviati e quindi elaborati per estrarre analisi che possono essere visualizzate dagli analisti.

L'implementazione della AnalyticsPipeline soluzione Game aiuta gli sviluppatori di giochi a lanciare una pipeline di dati serverless scalabile per importare, archiviare e analizzare i dati di telemetria generati da giochi e servizi. La soluzione supporta l'acquisizione di dati in streaming, consentendo agli utenti di ottenere informazioni dai giochi e da altre applicazioni in pochi minuti.

Per la telemetria personalizzata dei giochi, l'inserimento, l'archiviazione, l'elaborazione e l'analisi dei dati, offre AWS anche una serie di servizi specializzati per l'elaborazione e l'analisi dei big data.

Passaggi dell’implementazione

- Acquisisci dati di telemetria di gioco: raccogli dati sull'attività dei giocatori, sull'utilizzo del sistema, sulle eccezioni e sugli arresti anomali per comprendere le interazioni dei giocatori e identificare i problemi.
- Implementa la raccolta di dati di telemetria: utilizza funzionalità o posizioni di gioco predefinite per raccogliere dati di telemetria e inviarli ai servizi di backend, archiviandoli localmente se il backend non è raggiungibile.
- Utilizza soluzioni di analisi: utilizza AWS servizi come Game AWS Analytics Pipeline per l'acquisizione, l'archiviazione e l'analisi scalabili dei dati, oltre a servizi specializzati di elaborazione e analisi dei big data.

GAMEOPS06-BP02 Aggiorna e adatta l'approccio al test di carico man mano che il gioco cambia

L'ottimizzazione dell'approccio al load testing è un processo continuo che dovrebbe evolversi parallelamente al ciclo di sviluppo del gioco. Man mano che il gioco cresce in termini di complessità, base di utenti e set di funzionalità, la strategia di test di carico deve adattarsi per verificare che simuli accuratamente le condizioni del mondo reale e fornisca informazioni utili.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Considera i seguenti aspetti:

Scenari di test mancanti o obsoleti

Man mano che vengono aggiunte nuove funzionalità a un gioco durante il processo di sviluppo, crea ed esegui nuovi scenari di test di carico per convalidare le prestazioni e la scalabilità delle nuove funzionalità. Allo stesso modo, le caratteristiche e le funzionalità vengono spesso rifattorizzate per migliorare le prestazioni, rispondere al feedback dei giocatori o allinearsi ai nuovi obiettivi di progettazione, richiedendo l'aggiornamento continuo degli scenari di test per stare al passo con i cambiamenti e testare e riflettere realmente lo stato del sistema.

Nuovi framework per i test di carico

Gli sviluppatori potrebbero dover modificare i framework di test di carico per una serie di motivi:

- Il framework iniziale potrebbe non essere più in grado di simulare adeguatamente il carico dell'utente o di fornire il livello necessario di informazioni sulle prestazioni del sistema
- Le nuove funzionalità di gioco potrebbero richiedere il test di carico, il supporto per nuovi protocolli o APIs punti di integrazione
- Gli sviluppatori potrebbero desiderare funzionalità più avanzate man mano che acquisiscono maggiore dimestichezza con il processo di test di carico
- Preferenza per framework che meglio si allineano alle competenze tecniche del team, ai linguaggi di programmazione o alle toolchain esistenti

Valutando attentamente e adattandosi nel tempo, gli sviluppatori possono allineare il processo di load test alle mutevoli esigenze del gioco e continuare a fornire le informazioni necessarie per ottimizzare e migliorare l'esperienza utente complessiva.

Ottimizzazione dei costi

La facilità e la comodità di utilizzare AWS i servizi gestiti possono essere estremamente vantaggiose, soprattutto nelle prime fasi di sviluppo. Questi servizi riassumono la gestione dell'infrastruttura sottostante, consentendo ai team di configurare rapidamente la propria soluzione e concentrarsi esclusivamente sulla creazione di scenari di test di carico e sull'analisi dei risultati. Tuttavia, l'utilizzo dei servizi gestiti può spesso comportare costi più elevati a causa del valore aggiunto e della praticità che offrono, come il provisioning, la configurazione e la manutenzione dell'infrastruttura, oltre a fornire funzionalità di elevata disponibilità, scalabilità e monitoraggio.

Man mano che i team maturano e acquisiscono maggiore dimestichezza e confidenza con il processo di test di carico, può arrivare il momento in cui l'autogestione dell'infrastruttura può offrire ulteriori ottimizzazioni e risparmi sui costi. Sebbene questo approccio pratico aumenti il sovraccarico operativo, avere il controllo diretto sulle risorse di elaborazione, sulle configurazioni, sui comportamenti di scalabilità e sull'utilizzo delle risorse può sbloccare nuove opportunità di ottimizzazione e riduzione dei costi. Ad esempio, potrebbe essere opportuno che i team inizino il percorso di test di carico con un'architettura AWS Fargate serverless, per poi passare alla gestione automatica dei nodi sottostanti in un cluster Amazon EKS.

Passaggi dell’implementazione

- Aggiorna gli scenari di test: crea e aggiorna continuamente scenari di test di carico per convalidare nuove caratteristiche e funzionalità rifattorizzate e verificare che riflettano lo stato attuale del gioco.
- Valuta i framework di test di carico: adattati ai nuovi framework secondo necessità per simulare il carico degli utenti, supportare nuovi protocolli e allinearti alle competenze e alle toolchain del team.
- Ottimizzazione dei costi: inizia con i AWS servizi gestiti per semplificare e rendere più pratico l'utilizzo, quindi prendi in considerazione l'idea di gestire autonomamente l'infrastruttura per risparmiare sui costi man mano che il team acquisisce maggiore dimestichezza con il processo di test di carico.

Resources

Consulta le seguenti risorse per saperne di più sulle nostre best practice relative all'eccellenza operativa.

Documentazione e blog

- Le migliori pratiche di architettura per Game Tech

- Gestire Your Game Studio su AWS pt.1
- Gestire Your Game Studio su pt. AWS 2
- Gestire Your Game Studio su AWS pt. 3
- Creazione del tuo AWS ambiente di best practice
- strategia multi-account per la tua landing zone della Control Tower
- Game Analytics Pipeline
- Massimizza le informazioni sui dati di gioco con Game Analytics Pipeline
- Leveraging AWS Glue e Amazon Redshift Spectrum for Player Insights
- Come configurare una CI/CD pipeline su
- How Good Job Games accelera del 43% con AWS Build Pipeline
- Implementazione di una pipeline di compilazione per le app Unity Mobile
- Altri blog pertinenti CI/CD
- Gioco DevOps semplificato con il blog Game-Server CD Pipeline
- Harmony Games implementa un backend di gioco completamente personalizzato che utilizza () AWS Cloud Development Kit (AWS CDK)AWS CDK
- GameLiftPreparati per il lancio
- Hosting di server di gioco ibridi con Amazon Gamelift Anywhere
- Accelera lo sviluppo di server di gioco con Amazon Gamelift Anywhere e Amazon Gamelift Agent
- Come ospitare un gioco Unreal Engine per meno di $1 per giocatore con Amazon Gamelift
- Nuova guida alle soluzioni per creare backend di gioco scalabili e multipiattaforma su AWS
- Carica il test del motore di gioco di backend Pragma su 1 milione di utenti simultanei su AWS
- In che modo Code Wizards ha testato il carico di Nakama di Heroic Lab su due milioni di giocatori simultanei con AWS
- Le migliori pratiche con le unità organizzative
- AWS X-Ray
- AWS Conto alla rovescia
- AWS per Games Solutions Hub

Soluzioni dei partner

- New Relic

- APM Splunk
- Backtrace.io
- Sentinella
- Datadog APM
- Honeycomb.io

Whitepaper

- Organizzazione dell'ambiente utilizzando più account
- Introduzione ai modelli di sviluppo di giochi scalabili su AWS

Video

- YouTubeserie: Building Games on AWS
- AWS per giochi: Boss LEVEL Podcast
- Re:Invent 2023: ampliamento AWS per i primi 10 milioni di utenti
- Re:Invent 2022: In che modo Riot Games elabora 20 TB di analisi ogni giorno su AWS
- Re:Invent 2022: How AWS and Riot Games ha creato un motore di reporting sulla governance
- Re:Invent 2023: implementazione di modelli di progettazione distribuiti su AWS
- Re:Invent 2023: portare a milioni un gioco multiplayer con Mortal Kombat 1
- Re:Invent 2022: l'evoluzione dell'ingegneria del caos su Netflix
- Re:Invent 2023: migliori pratiche per la governance del cloud
- Re:Invent 2023: Le migliori pratiche per la creazione di architetture multiregionali su AWS

Materiali per la formazione

- Curriculum: Guida introduttiva a AWS For Games, parte 1

Sicurezza

Il pilastro della sicurezza include la capacità di proteggere informazioni, sistemi e asset fornendo al contempo valore aziendale attraverso la valutazione e la mitigazione del rischio. A causa della visibilità globale e del gran numero di giocatori, i giochi sono un obiettivo ideale per sfruttatori, hacker e chiunque sia alla ricerca di modi per sfruttare e abusare dei sistemi. Ciò può spesso comportare un'esperienza deludente per i giocatori e un aumento dei costi per lo sviluppatore del gioco se non si dispone di solide basi di sicurezza.

Come descritto nel modello di responsabilità condivisa, è importante capire quali aspetti della sicurezza ricadono sotto la responsabilità AWS e quali sono a carico del cliente, in modo da essere pronti a mantenere un elevato livello di sicurezza. Questo pilastro fornisce le migliori pratiche e linee guida sulla sicurezza del cloud da prendere in considerazione quando sviluppi e gestisci giochi nel cloud.

Prima di progettare un sistema, è necessario stabilire una serie di best practice di sicurezza che includano i controlli degli accessi. Inoltre, dovresti essere in grado di identificare gli incidenti di sicurezza e proteggere i tuoi sistemi e servizi, mantenendo al contempo la riservatezza e l'integrità dei dati attraverso la protezione dei dati. Dovresti avere dei processi ben definiti e rodati per rispondere a eventuali problemi di sicurezza. Questi strumenti e tecniche sono importanti perché supportano obiettivi aziendali come la prevenzione delle perdite finanziarie o il rispetto degli obblighi normativi.

Esempio del cliente

AnyCompany Games è uno studio di giochi immaginario che sta migliorando il proprio livello di sicurezza. La sicurezza può essere facile da capire quando c'è una spiegazione della sua applicazione diretta. AnyCompany I giochi vengono utilizzati in questa sezione per contestualizzare le migliori pratiche di sicurezza descritte nel pilastro

Aree di interesse

- Principi di progettazione
- Nozioni di base sulla sicurezza
- Sicurezza continua
- Gestione dell’identità e degli accessi
- Controllo accessi
- Rilevamento

- Protezione dell’infrastruttura
- Risposta agli incidenti
- Sicurezza delle applicazioni
- Automatizza la sicurezza
- Modellazione delle minacce
- Resources

Principi di progettazione

Oltre ai principi di progettazione del pilastro della sicurezza del white paper Well-Architected Framework, il seguente principio di progettazione può rafforzare la sicurezza del carico di lavoro di gioco nel cloud:

- Monitora e modera il comportamento di utilizzo dei giocatori: acquisisci e analizza i dati di utilizzo per capire come i giocatori interagiscono con le tue funzionalità di gioco e social. Analizzando questi dati, puoi rilevare e rispondere a comportamenti offensivi e inappropriati che possono peggiorare l'esperienza del giocatore.

Nozioni di base sulla sicurezza

GAMESEC01: Come implementate i fondamenti della sicurezza per lo sviluppo di giochi?

Gli studi di gioco richiedono un approccio alla sicurezza unico che protegga sia gli ambienti di sviluppo che i servizi per i giocatori dal vivo. Una solida strategia AWS di sicurezza per gli studi di gioco richiede tre componenti interconnessi: una struttura multi-account, un'autenticazione avanzata e una strategia di autorizzazione chiara che utilizzi le politiche IAM. Una AWS struttura multi-account consente agli studi di separare diversi progetti di gioco, fasi di sviluppo e ambienti di strumenti. Ciò offre agli studi un controllo più granulare su aspetti come l'accesso ad ambienti o servizi specifici. L'attivazione dell'autenticazione avanzata consente ai membri del team di accedere in modo sicuro alle risorse di sviluppo indipendentemente dal fatto che lavorino in studio o in remoto, mantenendo al contempo controlli rigorosi sul codice sorgente, sulle build dei giochi e sugli strumenti proprietari. Gli studi dovrebbero inoltre disporre di una chiara strategia di autorizzazione per la concessione delle autorizzazioni utilizzando il principio del privilegio minimo con autorizzazioni e ruoli IAM. Usa i

ruoli IAM per assegnare le autorizzazioni tra i diversi ruoli dei team di sviluppo, ad esempio dando ai team di sviluppo l'accesso a AWS servizi di basso livello e limitando al contempo artisti e designer a gestire risorse e sistemi di sviluppo specifici. Questo approccio specializzato verifica che gli studi di gioco siano in grado di proteggere la loro proprietà intellettuale, mantenere flussi di lavoro di sviluppo efficienti e scalare i propri team in modo sicuro, offrendo al contempo agli sviluppatori l'accesso appropriato per iterare rapidamente sui propri progetti.

Best practice

- GAMESEC01-BP01 Usa i ruoli e l'accesso federato, anziché l'utente root dell'account, per eseguire azioni sul tuo ambiente AWS
- GAMESEC01-BP02 AWS Control Tower Da utilizzare per configurare rapidamente un ambiente multi-account su AWS
- GAMESEC01-BP03 Utilizza politiche di ruolo con privilegi minimi adattate a funzioni lavorative specifiche
- GAMESEC01-BP04 Usa ruoli e politiche di accesso federato insieme a politiche di accesso a livello di account per concedere l'accesso alle tue risorse AWS
- GAMESEC01-BP05 Usa un provider di identità centrale

GAMESEC01-BP01 Usa i ruoli e l'accesso federato, anziché l'utente root dell'account, per eseguire azioni sul tuo ambiente AWS

La prima volta che si crea un account Account AWS, si inizia con un'identità nota come utente root, a cui si accede utilizzando l'indirizzo e-mail e la password associati all'account. L'utente root ha accesso completo ai AWS servizi e alle risorse all'interno di quell'account. Nella maggior parte dei casi, è consigliabile evitare di utilizzare l'utente root per day-to-day le attività. Se è richiesto l'accesso a livello di root, verificate che sia assolutamente necessario e verificate che siano presenti registri e barriere aggiuntivi per monitorarne l'utilizzo.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

In una AWS Organizations configurazione, ogni account ha ancora il proprio utente root, ma l' day-to-dayaccesso dovrebbe invece essere gestito tramite i ruoli IAM e gli utenti di IAM Identity Center. Crea un accesso basato sui ruoli personalizzato in base alle fasi e ai team del ciclo di vita del tuo gioco. Ad esempio, il team addetto alle operazioni live potrebbe aver bisogno di autorizzazioni per gestire gli

eventi di gioco, mentre gli sviluppatori devono poter accedere agli aggiornamenti push. Quando lavori con servizi o partner di terze parti, utilizza l'accesso federato per consentire una collaborazione sicura senza esporre l'infrastruttura sensibile. Questo approccio verifica che ogni utente o partner disponga solo dell'accesso di cui ha bisogno, mantenendo al contempo la sicurezza dell'infrastruttura di gioco e dei dati dei giocatori.

Esempio del cliente

AnyCompany Games ha implementato il controllo degli accessi basato sui ruoli durante lo sviluppo del nuovo gioco. Utilizzando ruoli IAM specifici per i loro diversi team di sviluppo, evitano di utilizzare credenziali condivise. Questa configurazione consente a un team di sviluppo di assumere un ruolo per i sistemi di gioco principali, mentre il ruolo del team addetto ai contenuti è quello di accedere solo ai servizi di gestione delle risorse.

Passaggi dell'implementazione

- Non utilizzate l'utente root dopo aver configurato un account a meno che non sia assolutamente necessario. Crea l'account, proteggi l'utente root e crea immediatamente i ruoli IAM di amministrazione richiesti e assegna quel ruolo all'utente federato.
- Usa l'utente root solo quando devi eseguire un numero limitato di attività disponibili solo per l'utente root. Esempi di queste attività includono la modifica dell'indirizzo e-mail dell'utente root e la modifica del piano di AWS supporto.

GAMESEC01-BP02 AWS Control Tower Da utilizzare per configurare rapidamente un ambiente multi-account su AWS

Se inizi a utilizzarlo AWS con un solo account, potresti scoprire che il tuo studio di gioco ne sta crescendo man mano che il processo di sviluppo del gioco avanza. Ad esempio, utilizzandone uno Account AWS, potresti iniziare a raggiungere i limiti di servizio oppure i costi per diversi progetti e carichi di lavoro potrebbero diventare più complessi. La creazione di account diversi per titoli e ambienti di gioco diversi consente ai team di sperimentare nuove funzionalità, aggirare i limiti del servizio e mantenere il livello di sicurezza e la conformità. Implementando una strategia multi-account in AWS, puoi trarre vantaggio dalla distribuzione dei limiti di servizio su più account e ottenere informazioni dettagliate sui costi. AWS

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

È un malinteso comune che l'utilizzo di più utenti crei Account AWS automaticamente più confusione e richieda più tempo. Piuttosto, l'utilizzo di AWS servizi progettati per facilitare la gestione di più account può aiutare lo studio di gioco a dedicare meno tempo alla gestione degli account.

Puoi utilizzare AWS Control Tower questo servizio per fornire in modo sicuro un ambiente con più account AWS . Control Tower è consigliato se stai costruendo un nuovo AWS ambiente, stai iniziando il tuo viaggio o sei completamente nuovo AWS. AWS Durante il breve processo di configurazione, puoi integrarti con altri AWS servizi coinvolti nella gestione degli account e dell'accesso degli utenti AWS Organizations, come Service Catalog e AWS IAM Identity Center.

Esempio del cliente

AnyCompany I giochi inizialmente funzionavano da un unico Account AWS sistema, ma hanno incontrato diversi ostacoli quando uno dei team di sviluppo dei giochi ha raggiunto i limiti di EC2 servizio durante un beta test cruciale. Allo stesso tempo, il team di sviluppo di un altro gioco aveva difficoltà nell'allocazione delle risorse per la pipeline di test automatizzata. La situazione raggiunse un punto di svolta quando AnyCompany Games non riuscì a separare con precisione i costi tra i progetti, rendendo difficile la definizione del budget per lo sviluppo di ciascun gioco.

AnyCompany Games ha quindi implementato una strategia multi-account utilizzando AWS Control Tower. Hanno creato account separati per ogni progetto di gioco, con ambienti di sviluppo, controllo qualità e produzione distinti. Questa separazione a livello di account isola i dati e le risorse di ogni progetto, in modo che i team che lavorano su un gioco non possano accedere o modificare le risorse di un altro. AWS Organizations In seguito, hanno stabilito una struttura di fatturazione centralizzata che mostrava chiaramente i costi di infrastruttura di ogni gioco e ha anche creato politiche di accesso a livello di organizzazione.

Passaggi dell’implementazione

- Usa AWS Control tower per configurare un ambiente automatizzato con più account.
- Organizza gli account in base agli ambienti (come sviluppo, controllo qualità e produzione).
- Utilizza AWS IAM Identity Center and Service Catalog per centralizzare le autorizzazioni degli utenti e semplificare il provisioning delle risorse tra gli account.

GAMESEC01-BP03 Utilizza politiche di ruolo con privilegi minimi adattate a funzioni lavorative specifiche

La configurazione delle policy IAM è una parte essenziale per stabilire una solida base di sicurezza. Quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Ad esempio, i team addetti al controllo qualità devono poter apportare modifiche negli ambienti di test, ma non dovrebbero avere la possibilità di modificare l'ambiente di produzione.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Potresti iniziare con autorizzazioni ampie, come le politiche gestite, mentre esplori le autorizzazioni necessarie per il tuo carico di lavoro o il tuo caso d'uso. Man mano che il tuo caso d'uso matura, puoi lavorare per ridurre le autorizzazioni concesse per lavorare con il privilegio minimo.

Passaggi dell’implementazione

- Segui la pratica delle autorizzazioni con privilegi minimi per creare ruoli IAM per utenti e applicazioni.
- Utilizza le policy AWS gestite per fornire rapidamente un ampio accesso, identificando al contempo le autorizzazioni specifiche di cui i team o le applicazioni hanno bisogno per svolgere le proprie attività.
- Gli studi possono anche utilizzare la generazione di policy di IAM Access Analyzer per generare policy IAM personalizzate basate su CloudTrail eventi che identificano le azioni e i servizi utilizzati da una voce IAM.
- Esamina regolarmente le policy IAM e modifica le policy eccessivamente permissive.

GAMESEC01-BP04 Usa ruoli e politiche di accesso federato insieme a politiche di accesso a livello di account per concedere l'accesso alle tue risorse AWS

I nuovi AWS utenti spesso utilizzano le policy IAM solo quando concedono l'accesso ad altri. Tuttavia, se lo utilizzi AWS Organizations, valuta come utilizzare le policy di controllo del servizio insieme alle policy IAM per garantire ai membri del team di studio e agli appaltatori i livelli di accesso necessari.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Puoi creare policy IAM per consentire o negare l'accesso ai AWS servizi o alle azioni API con cui funziona. AWS Identity and Access Management Possono essere applicate solo alle identità IAM, come utenti, gruppi o ruoli. Ad esempio, una policy IAM potrebbe essere utilizzata per fornire a un utente l'accesso in sola lettura ad Amazon S3.

Le policy di controllo dei servizi (SCPs) sono dei punti di riferimento per te. Account AWS Un SCP non concede autorizzazioni, ma vengono utilizzate per limitare le azioni sui AWS servizi per gli account dei singoli membri. Ad esempio, un SCP può negare a un uomo di accedere Account AWS a una particolare regione.

Quando viene intrapresa un'azione, la politica IAM pertinente viene valutata in combinazione con. SCPs Seguendo l'esempio precedente, se un ruolo tenta di eseguire un' EC2 istanza, IAM indica se è consentito («Allow» per ec2:RunInstances) e SCPs determinerà se la scelta della regione è valida («us-east-1" è consentito, ma «us-west-1" è negato da SCP).

Stratificando le politiche IAM, SCPs puoi verificare che chiunque acceda alle tue AWS risorse riceva solo le autorizzazioni appropriate di cui ha bisogno. Ciò è particolarmente importante da considerare se le tue Account AWS risorse si estendono su più regioni, ma non tutti i membri del tuo studio di gioco devono accedervi tutte.

Puoi personalizzare le policy IAM per concedere a team specifici autorizzazioni specifiche per l'aggiornamento delle configurazioni di gioco, la gestione dei dati dei giocatori, la configurazione di eventi promozionali e la moderazione dei contenuti generati dagli utenti. Nel frattempo, puoi usarli SCPs per far rispettare i controlli a livello di organizzazione fondamentali per le operazioni di gioco. Questi potrebbero includere la limitazione dell'implementazione solo nelle regioni approvate in cui il gioco opera, la prevenzione dell'accesso non autorizzato agli archivi di dati sensibili dei giocatori, l'applicazione dei requisiti di conformità e il controllo dei costi limitando l'utilizzo del servizio tra gli account di sviluppo.

Passaggi dell’implementazione

- Utilizza le policy IAM per gestire le autorizzazioni per singoli utenti, gruppi o ruoli.
- Utilizza le policy di controllo del servizio (SCPs) AWS Organizations per applicare le autorizzazioni a livello di account.
- Combina le policy IAM e SCPs concedi solo l'accesso richiesto a utenti e account specifici.

Resources

- Politiche e autorizzazioni in AWS IAM
- Policy di controllo dei servizi
- AWS managed policies for job functions

GAMESEC01-BP05 Usa un provider di identità centrale

Un provider di identità centrale funge da unica fonte per l'archiviazione e la gestione delle credenziali, delle identità, delle autorizzazioni e dell'autenticazione degli utenti.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Utilizza un provider di identità centralizzato per semplificare il processo di autenticazione degli utenti, applicare politiche di sicurezza coerenti e semplificare la gestione degli utenti tra le tue applicazioni. Account AWS Un approccio centralizzato elimina la necessità di gestire le identità e le credenziali degli utenti separatamente, il che riduce il rischio di incongruenze, ridondanze e altre vulnerabilità di sicurezza. Il consolidamento delle identità e dell'autenticazione degli utenti in un unico posto consente inoltre una migliore visibilità, controllo e verificabilità per l'intero ambiente. AWS

Esempio del cliente

AnyCompany Games ha dovuto affrontare sfide significative nella gestione dell'accesso degli sviluppatori attraverso la loro infrastruttura in rapida espansione. AWS Il team di sviluppo è passato da 50 a 200 persone per tre titoli principali. Inizialmente, ogni team di progetto gestiva le proprie credenziali di AWS accesso, con conseguenti pratiche di sicurezza incoerenti, ritardi nell'onboarding dei nuovi sviluppatori e occasionali incidenti di sicurezza.

Lo studio ha implementato AWS IAM Identity Center come provider di identità centrale, consolidando la gestione degli utenti in un unico sistema. L'hanno collegato alla loro directory aziendale esistente, consentendo agli sviluppatori di utilizzare le stesse credenziali aziendali per AWS l'accesso. Ora gli sviluppatori utilizzano il loro unico login aziendale esistente per ottenere l' AWS accesso necessario per completare il proprio lavoro

Passaggi dell’implementazione

- Prendi in considerazione l'utilizzo di AWS IAM Identity Center come provider di identità centrale. Ciò consente una gestione coerente degli accessi in tutta l'azienda Account AWS, fornisce ai

dipendenti l'autenticazione Single Sign-On e semplifica il controllo degli accessi degli utenti alle applicazioni. AWS IAM Identity Center si collega anche alle identità aziendali esistenti dei provider di identità supportati.

Sicurezza continua

GAMESEC02: Come si ottengono, mantengono e monitorano le migliori pratiche di sicurezza continue?

L'adesione alle migliori pratiche di sicurezza è fondamentale per le aziende di tutti i settori, ma soprattutto nel settore dei giochi. L'industria dei giochi si basa sulla promozione e sul mantenimento della fiducia dei giocatori e sulla creazione di una solida reputazione, e anche piccoli problemi di sicurezza possono minare rapidamente tale fiducia.

Inoltre, la natura globale del settore dei giochi richiede il rispetto di varie normative e standard di settore che disciplinano la protezione dei dati, la privacy dei consumatori e la sicurezza nelle regioni in cui vengono offerti i giochi. Un gameplay equo e sicuro è un altro aspetto fondamentale che sottolinea l'importanza di solide misure di sicurezza. L'imbroglio, la pirateria informatica e altre forme di sfruttamento del gioco possono compromettere l'esperienza di gioco dei giocatori legittimi, il che rende essenziali controlli di sicurezza rigorosi per mantenere l'integrità del gioco e promuovere condizioni di parità per i partecipanti.

Best practice

- GAMESEC02-BP01 Usa modelli pronti per l'implementazione per pratiche di sicurezza standard
- GAMESEC02-BP02 Utilizza tecniche di riparazione automatizzate quando si verifica un evento di sicurezza

GAMESEC02-BP01 Usa modelli pronti per l'implementazione per pratiche di sicurezza standard

Ready-to-deployi modelli forniscono un modo proattivo e agile per valutare il livello di sicurezza nel cloud. I modelli preconfigurati valutano la sicurezza del cloud e implementano tempestivamente le modifiche necessarie. I modelli comprendono un'ampia gamma di best practice relative a varie tecnologie e framework di sicurezza ampiamente accettati. L'utilizzo dei modelli può aiutare gli studi

di gioco a mantenere configurazioni di infrastruttura coerenti, soprattutto perché possono scalare e aggiungerne altre Account AWS per supportare nuovi carichi di lavoro.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Utilizzando AWS servizi e implementando ready-to-deploy modelli, gli sviluppatori di giochi possono valutare e rafforzare in modo proattivo la propria posizione di sicurezza sul cloud, salvaguardare la proprietà intellettuale, proteggere i dati dei giocatori e promuovere un panorama di gioco sicuro attraverso valutazioni di sicurezza regolari e un monitoraggio continuo per identificare e risolvere tempestivamente le potenziali vulnerabilità.

Esempio del cliente

AnyCompany Games ha dovuto affrontare una sfida importante nel prepararsi a lanciare il suo prossimo titolo nel settore europeo. Si sono resi conto che le loro pratiche di gestione dei dati esistenti non soddisfacevano i requisiti del GDPR. Si sono rivolti a AWS Security Hub CSPM and AWS Config and ai suoi ready-to-deploy modelli per trovare una soluzione. Il team ha implementato il pacchetto di conformità specifico per il GDPR AWS Config, che ha valutato automaticamente l'infrastruttura esistente rispetto agli standard GDPR. Questa scansione iniziale ha rivelato diverse lacune critiche, come politiche di conservazione dei dati improprie e controlli di accesso inadeguati su dove erano archiviati i dati dei giocatori. Utilizzando le regole predefinite del modello, AnyCompany Games ha rapidamente implementato le modifiche necessarie. Inoltre, i continui controlli automatici di conformità forniti dal modello hanno consentito al piccolo team di mantenere la conformità al GDPR senza problemi, anche se continuava ad aggiornare ed espandere il gioco.

Passaggi dell'implementazione

- Utilizza modelli per pratiche di sicurezza standard, come regole gestite e pacchetti di conformità e standard in AWS Config . AWS Security Hub CSPM
- Esamina i dettagli degli standard CSPM di Security Hub per determinare quali si adattano maggiormente alle esigenze di sicurezza del tuo studio di gioco.

GAMESEC02-BP02 Utilizza tecniche di riparazione automatizzate quando si verifica un evento di sicurezza

Utilizzando tecniche di riparazione automatizzate, gli sviluppatori di giochi possono proteggere e mantenere in modo proattivo la propria infrastruttura di gioco e ridurre al minimo il potenziale impatto

che un incidente di sicurezza potrebbe avere. Se viene rilevato un problema di sicurezza, utilizza un runbook per guidare la tua risposta alla situazione. Automatizza queste risposte laddove possibile per risolvere i problemi più rapidamente e ridurne l'impatto. Ciò migliora l'esperienza del giocatore riducendo la possibilità di tempi di inattività e interruzioni del gioco.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Prepararsi a rispondere ai problemi di sicurezza non solo salvaguarda l'esperienza dei giocatori, ma anche il rispetto dei vari standard normativi e di conformità. Inoltre, l'utilizzo di risposte di sicurezza automatizzate ridimensiona le operazioni di sicurezza man mano che i carichi di lavoro si espandono. AWS fornisce servizi per aiutare a identificare e automatizzare la risposta a questi incidenti.

Esempio del cliente

AnyCompany Games ha subito un grave incidente di sicurezza quando un bucket S3 contenente texture e modelli di personaggi inediti per il loro prossimo gioco è stato reso pubblico per errore durante un aggiornamento di routine della pipeline degli asset. Il sistema di sicurezza automatizzato ha rilevato la modifica dell'autorizzazione del bucket entro pochi minuti dalla modifica. Il sistema ha eseguito immediatamente la procedura di correzione: ripristino dello stato privato del bucket, registrazione dei tentativi di accesso durante la finestra di esposizione, notifica al team di sicurezza e creazione di un registro dettagliato delle modifiche alle autorizzazioni. CloudTrail

Passaggi dell'implementazione

- Utilizza la AWS soluzione Automated Security Response on per implementare i runbook di automazione che definiscono le azioni che verranno intraprese automaticamente in risposta agli eventi di sicurezza in. AWS Security Hub CSPM

Resources

- AWS for Games Blog — Gestire il tuo Game Studio su AWS: Parte 1
- AWS for Games Blog — Gestire il proprio Game Studio (AWS parte 2)
- Registra un'unità organizzativa esistente con AWS Control Tower
- Account AWS utente root
- Attività che richiedono credenziali utente root
- Risposta di sicurezza automatizzata attiva AWS

Gestione dell’identità e degli accessi

GAMESEC03: Come gestisci l'identità dei giocatori e la gestione degli accessi?

Quando sviluppi un gioco, devi stabilire in che modo fornire ai giocatori l'accesso al gioco e ai servizi correlati. La sezione seguente esplora le considerazioni di progettazione, tra cui l'autenticazione dei giocatori, l'autorizzazione e l'autenticazione a più fattori.

Best practice

- GAMESEC03-BP01 Determina il tuo approccio per identificare e controllare l'accesso dei giocatori all'ambiente e alle risorse del tuo gioco
- GAMESEC03-BP02 Autentica le richieste inviate al tuo servizio di backend di gioco
- GAMESEC03-BP03 Usa il tuo servizio di backend di gioco per convalidare le richieste dei giocatori di partecipare a una partita multiplayer
- GAMESEC03-BP04 Applica una rigorosa politica di sicurezza per gli account utente dei giocatori richiedendo una password complessa
- GAMESEC03-BP05 Fornisci ai giocatori la possibilità di configurare l'autenticazione a più fattori (MFA) sui propri account

GAMESEC03-BP01 Determina il tuo approccio per identificare e controllare l'accesso dei giocatori all'ambiente e alle risorse del tuo gioco

Questa decisione è influenzata dalla strategia di acquisizione e monetizzazione dei giocatori, dall'esperienza del giocatore e da altri fattori, come le funzionalità esistenti che potrebbero essere fornite dai partner di pubblicazione dei giochi. Ad esempio, un gioco potrebbe richiedere acquisti e richiedere al giocatore di creare un profilo utente per associare metodi di pagamento con denaro reale al proprio account.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

In alternativa, un gioco potrebbe voler ridurre la barriera all'ingresso per chi gioca per la prima volta eliminando la necessità di creare un account utente prima di giocare, aumentando così le

possibilità che un giocatore provi il gioco per la prima volta. In genere, i giochi implementano una o più combinazioni di identità del giocatore e approcci di gestione degli accessi per il loro gioco.

Accesso non autenticato o anonimo

Questo livello di accesso è utile in situazioni in cui un gioco non richiede al giocatore di creare un nuovo account utente o di collegarsi alla propria identità sui social network e sui sistemi di gioco. Questo è il modo più semplice e veloce per un giocatore di iniziare a giocare ed è particolarmente utile nei giochi per dispositivi mobili in cui uno sviluppatore di giochi potrebbe voler ridurre la barriera all'ingresso per l'esperienza iniziale.

In questo scenario di accesso, se desideri identificare l'utilizzo dall'installazione del gioco, puoi programmare il client di gioco in modo che generi e memorizzi un identificatore univoco sul dispositivo del giocatore. Questo identificatore univoco viene utilizzato per identificare il giocatore durante le sessioni di gioco sul proprio dispositivo e consentire report analitici sull'utilizzo nel tempo. Successivamente, se un giocatore sceglie di creare un account, puoi associare il nuovo account utente all'identificatore univoco generato in precedenza. Ciò collegherà la loro nuova identità di giocatore al loro utilizzo storico, che potrebbe includere statistiche e risultati di gioco.

Se alla fine un giocatore non crea e non collega un account, il dispositivo che il giocatore utilizza per interagire con il gioco può essere identificato in modo univoco, ma le informazioni recuperabili sul giocatore non vengono raccolte e archiviate. Pertanto, se il giocatore si rompe o perde il proprio dispositivo, anche i dati memorizzati in precedenza associati al dispositivo vengono persi e potrebbero non essere recuperabili.

Autenticazione con nome utente e password

Un gioco può consentire ai giocatori di creare i propri account utente con un nome utente e una password memorizzati nel backend del gioco. Ciò può verificarsi quando uno sviluppatore di giochi collabora con un editore di giochi che dispone già di un sistema di account giocatore con cui lo sviluppatore può integrarsi. In alternativa, uno sviluppatore che pubblica i propri giochi potrebbe voler semplificare l'esperienza del giocatore consentendogli di creare un unico account utente per l'accesso a tutti i giochi che pubblica.

Autenticazione e collegamento dell'account a social network e sistemi di gioco di terze parti

È comune che i giochi online e i giochi con funzionalità social offrano una federazione di provider di identità di terze parti per semplificare l'esperienza dei giocatori. Invece di chiedere ai giocatori di creare una combinazione di nome utente e password per l'autenticazione, puoi utilizzare la

federazione delle identità per consentire ai giocatori di autenticarsi utilizzando i propri account di terze parti sui social network e sui sistemi di gioco. Questa procedura di accesso semplifica l'esperienza di accesso e registrazione per i giocatori. Fornisce inoltre una comoda alternativa alla creazione obbligatoria dell'account e un metodo semplice per i giocatori di accedere ai giochi.

Per gli sviluppatori di giochi, una procedura di accesso federata può offrire un flusso di lavoro semplificato per la verifica dei giocatori. Può anche fornire un modo più affidabile per gestire i dati dei giocatori utilizzati per la personalizzazione. Questo perché non è necessario chiedere ai giocatori di fornirti determinati dati che probabilmente hanno già fornito al provider di identità terzo. Inoltre, questi sistemi forniscono l'integrazione con funzionalità social aggiuntive come la possibilità di collegare i giocatori con i loro amici.

Passaggi dell’implementazione

- Utilizza l'accesso non autenticato o anonimo per ridurre le barriere per i giocatori alle prime armi generando un identificatore univoco del dispositivo per tracciare l'utilizzo e abilitando il collegamento dell'account in un secondo momento.
- Implementa l'autenticazione con nome utente e password per account utente dedicati, utilizzando i sistemi di account giocatore esistenti o creando un'esperienza unificata tra i giochi.
- Integra provider di identità di terze parti per l'autenticazione federata, semplificando i processi di accesso e abilitando l'accesso alle funzionalità social e ai dati di personalizzazione.

GAMESEC03-BP02 Autentica le richieste inviate al tuo servizio di backend di gioco

L'autenticazione delle richieste inviate al servizio di backend di gioco può impedire che le richieste indesiderate abbiano successo.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

È necessario fornire un servizio di autenticazione per consentire ai giocatori di accedere, che restituisca token sicuri di breve durata, come un JSON Web Token (JWT), al client di gioco quando un giocatore si autentica con successo.

Questi token possono includere asserzioni relative alle rivendicazioni che contengono gli attributi del giocatore e altri metadati pertinenti. Questi metadati pertinenti possono essere utilizzati nelle richieste

successive inviate dal client di gioco al backend di gioco per autenticare le richieste e autorizzarle nel contesto del giocatore autenticato.

Hai la possibilità di progettare e creare il tuo sistema di autenticazione dei giocatori, che richiederebbe miglioramenti e manutenzione continui, oppure puoi utilizzare le funzionalità scalabili e sicure di registrazione, accesso e controllo degli accessi degli utenti fornite da Amazon Cognito.

I pool di utenti di Amazon Cognito includono una directory di utenti per l'autenticazione e l'autorizzazione. Un pool di utenti APIs ti consente di integrarlo nel gioco per i flussi di lavoro di registrazione, accesso e reimpostazione della password, che possono essere integrati con provider di identità di terze parti. Application Load Balancers e Amazon API Gateway forniscono entrambi integrazioni con Cognito per integrare l'autenticazione degli utenti per le richieste inviate ai backend di gioco personalizzati ospitati con questi servizi.

Se il tuo gioco supporta l'accesso anonimo e non riesci ad autenticare un giocatore, puoi utilizzare un approccio di autenticazione client per fornire un'esperienza più sicura durante l'integrazione con il backend di gioco. Se il client di gioco utilizza direttamente i AWS servizi, le richieste a tali servizi devono essere firmate utilizzando le credenziali. Per fornire credenziali al tuo client di gioco per utenti non autenticati, puoi utilizzare l' AWS SDK per recuperare credenziali di breve durata dai pool di identità di Amazon Cognito che possono essere utilizzate per firmare le tue richieste ai servizi. AWS Queste credenziali possono essere aggiornate dal tuo client di gioco.

Oltre all'integrazione diretta con l' AWS SDK dal client di gioco, puoi anche creare il tuo backend di gioco, utilizzando un servizio come Amazon API Gateway, che supporta l'autorizzazione personalizzata. Progettando il tuo servizio di backend di gioco, puoi ottenere un controllo autorevole sulle richieste con una logica lato server personalizzata.

Per ulteriori informazioni sulla creazione di un servizio di backend per i giochi ospitati tramite Amazon GameLift, consulta Progetta il tuo servizio client di gioco.

Esempio del cliente

AnyCompany Games ha migliorato la sicurezza del prossimo titolo adottando un approccio di autenticazione e autorizzazione gestite. Invece di mantenere un sistema di nome utente e password personalizzato, hanno utilizzato i pool di utenti di Amazon Cognito per gestire la registrazione e l'accesso dei giocatori e i pool di identità per supportare l'accesso anonimo ai giocatori che provavano la modalità allenamento prima di creare un account. Hanno inoltre implementato una logica di autorizzazione personalizzata all'interno del gioco per riconoscere i ruoli di amministratore definiti in Cognito, garantendo a tali utenti l'accesso a speciali funzionalità di gestione del gioco.

Passaggi dell’implementazione

- Usa i pool di utenti di Amazon Cognito per gestire l'autenticazione con token sicuri come l'abilitazione di funzionalità come registrazione JWTs, accesso e reimpostazione delle password.
- Recupera credenziali di breve durata dai pool di identità di Amazon Cognito per consentire agli utenti anonimi di interagire in modo sicuro con i servizi. AWS
- Implementa backend di gioco personalizzati utilizzando Amazon API Gateway per una logica di autenticazione lato server personalizzata.

GAMESEC03-BP03 Usa il tuo servizio di backend di gioco per convalidare le richieste dei giocatori di partecipare a una partita multiplayer

In genere, nelle partite multiplayer, un giocatore partecipa a una sessione di gioco selezionando un'opzione direttamente da un elenco di sessioni disponibili, oppure invia una richiesta per trovare una partita. Quest'ultimo approccio attribuisce allo sviluppatore del gioco la responsabilità di individuare una sessione di gioco idonea e di fornire le informazioni di connessione (di solito un indirizzo IP e un numero di porta) al client di gioco del giocatore. L'implementazione può variare a seconda del genere di gioco che stai sviluppando, ma in ogni caso, è una buona pratica di sicurezza eseguire la convalida lato server della richiesta di iscrizione di un giocatore a una partita.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Ad esempio, in una partita multigiocatore basata su sessioni, la richiesta di un giocatore di partecipare a una sessione di gioco deve essere convalidata dal software del server di gioco con il servizio di matchmaking di backend di gioco prima di autorizzare la connessione al server. Quando un giocatore richiede di partecipare a una sessione di gioco, il server di gioco dovrebbe verificare la richiesta di un identificatore univoco, come un ID di sessione del giocatore e un ticket generato dal server che era stato precedentemente fornito al client di gioco dal tuo servizio di matchmaking di backend di gioco.

Dopo aver avviato la connessione al server di gioco, il software lato server può utilizzare queste informazioni per verificare con il servizio di matchmaking che la richiesta di connessione del giocatore sia valida e verificare che il giocatore non si stia unendo a un posto precedentemente riservato nella sessione di gioco a un altro giocatore.

Per i giochi ospitati su Amazon GameLift, consulta client/server Interazioni di gioco con Amazon GameLift Servers per un esempio di come può essere implementato questo tipo di convalida lato server.

Esempio del cliente

Durante il lancio iniziale AnyCompany della versione beta di Games, hanno scoperto che i giocatori aggiravano il loro sistema di matchmaking collegandosi direttamente ai server di gioco, causando seri problemi di integrità competitiva. Quando i giocatori di alto livello hanno scoperto di poter condividere gli indirizzi IP dei server con gli amici, hanno iniziato a eludere il sistema di matchmaking basato sulle abilità, con il risultato che i giocatori esperti si univano alle partite alle prime armi e creavano un'esperienza frustrante per i nuovi giocatori. AnyCompany Games ha risposto implementando un sistema di convalida lato server che generava ticket di sessione unici per ogni richiesta di matchmaking. Il sistema richiedeva sia i ticket di richiesta del giocatore che quelli del matchmaking IDs e verificava i tentativi di connessione con il servizio di matchmaking di backend.

Passaggi dell'implementazione

- Convalida le richieste di iscrizione dei giocatori sul lato server utilizzando identificatori univoci come la sessione del giocatore e i ticket generati dal server. IDs
- Conferma la validità delle richieste di connessione con il servizio di matchmaking per bloccare l'accesso non autorizzato.
- Verifica che i posti riservati nelle sessioni di gioco non siano accessibili a giocatori non autorizzati durante il processo di convalida.

GAMESEC03-BP04 Applica una rigorosa politica di sicurezza per gli account utente dei giocatori richiedendo una password complessa

Se un gioco offre ai giocatori la possibilità di creare un account utente con una password, è necessario richiedere che le password dei giocatori aderiscano a regole rigorose. Ad esempio, i pool di utenti di Amazon Cognito offrono la possibilità di definire i requisiti di password per gli account utente. Stabilire una politica rigorosa in materia di password può proteggere gli account dei tuoi giocatori dall'invasione dovuta all'ingegneria sociale e agli attacchi di forza bruta.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Esempio del cliente

AnyCompany I giochi hanno subito una crisi quando il loro popolare titolo ha subito un'ondata di dirottamenti di account a causa delle deboli politiche in materia di password. I giocatori che utilizzavano password semplici come «password123" stavano diventando vittime di attacchi automatici di forza bruta, con conseguente perdita di oggetti e compromissione della valuta di gioco. Per ovviare a questo problema, AnyCompany Games ha rinnovato il proprio sistema di accesso e ha imposto che le password, non utilizzate in precedenza, includessero almeno una lettera maiuscola, un numero, un carattere speciale e una lunghezza minima di 15 caratteri.

Passaggi dell’implementazione

- Richiedi politiche di password complesse per gli account dei giocatori per migliorare la sicurezza.
- Usa i pool di utenti di Amazon Cognito per definire e applicare i requisiti relativi alle password.

GAMESEC03-BP05 Fornisci ai giocatori la possibilità di configurare l'autenticazione a più fattori (MFA) sui propri account

Gli account dei giocatori possono essere una risorsa per i malintenzionati, in particolare nei giochi che supportano la valuta e gli acquisti in-game. A causa della pervasività degli attacchi di hacking sugli account dei giocatori e di ingegneria sociale, offri ai giocatori la possibilità di migliorare la sicurezza dei propri account configurando l'autenticazione a più fattori (MFA).

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Quando un giocatore tenta di accedere al proprio account utilizzando l'MFA, viene inviato un codice temporaneo al proprio indirizzo e-mail, numero di telefono o a un'app mobile di autenticazione a più fattori appositamente creata. Per autenticarsi correttamente, il giocatore deve quindi inserire il codice nel sistema di accesso entro un periodo di tempo limitato.

La MFA può essere utilizzata anche per proteggere gli account che stanno tentando di autenticarsi da una nuova geolocalizzazione, gli account che sono stati segnalati dall'assistenza giocatori per potenziali attività dannose e persino gli account che non hanno effettuato l'accesso al gioco per un periodo prolungato.

Ad esempio, i pool di utenti di Amazon Cognito possono configurare l'autenticazione a più fattori nelle directory degli utenti.

Passaggi dell'implementazione

- Abilita l'autenticazione a più fattori (MFA) per migliorare la sicurezza dell'account giocatore.
- Utilizza codici temporanei inviati tramite e-mail, telefono o app MFA per verificare l'accesso all'account.
- Applica la MFA per nuove geolocalizzazioni, account contrassegnati o account con inattività prolungata.

Controllo accessi

GAMESEC04: Come si blocca l'accesso non autorizzato ai contenuti di gioco?

I giochi moderni includono una notevole quantità di contenuti, come i contenuti scaricabili (DLC), che rappresentano un aspetto importante del coinvolgimento dei giocatori e della monetizzazione del gioco. I giocatori si aspettano un flusso continuo di nuovi personaggi, livelli e sfide, che richiedono agli sviluppatori di giochi di stare al passo con una domanda costante di nuovi contenuti per fidelizzare i giocatori. La varietà e la dimensione dei contenuti possono variare notevolmente a seconda del tipo di gioco e del dispositivo su cui viene giocato. Indipendentemente dal sistema di gioco, proteggi i contenuti del gioco da accessi non autorizzati.

Best practice

- GAMESEC04-BP01 Limita l'accesso ai contenuti scaricabili a client e utenti autorizzati
- GAMESEC04-BP02 Limita l'accesso all'origine alle reti di distribuzione dei contenuti autorizzate () CDNs
- GAMESEC04-BP03 Implementa restrizioni geografiche per limitare l'accesso non autorizzato
- GAMESEC04-BP04 Limita l'accesso ai contenuti con soluzioni di gestione dei diritti digitali (DRM)

GAMESEC04-BP01 Limita l'accesso ai contenuti scaricabili a client e utenti autorizzati

Limita l'accesso ai contenuti di gioco da parte di applicazioni e client autorizzati. Prendi in considerazione l'idea di utilizzare Amazon S3 come origine economica e scalabile per l'archiviazione di contenuti di gioco scaricabili e CloudFront Amazon per fornire ai giocatori una distribuzione di

contenuti performante a livello globale. Entrambi i servizi forniscono meccanismi integrati per limitare l'accesso ai dati archiviati, ad esempio l'accesso agli utenti autenticati.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Concessione dell'accesso ai contenuti archiviati in Amazon S3

Quando devi concedere l'accesso ai contenuti archiviati in S3, ci sono diversi fattori da considerare. Per impostazione predefinita, solo chi Account AWS ha creato un bucket S3 può accedere agli oggetti memorizzati al suo interno. Per concedere l'accesso alle tue applicazioni interne e gestire i contenuti archiviati nei bucket Amazon S3, usa AWS Identity and Access Management (IAM) per creare policy che forniscano un accesso appropriato.

I ruoli IAM possono essere associati a utenti, sistemi o applicazioni federati ospitati in servizi, come Amazon EC2 AWS Lambda, e applicazioni basate su contenitori ospitate in Amazon EKS e Amazon ECS. Ad esempio, puoi utilizzare l' AWS SDK o pubblicare e gestire risorse di contenuti AWS CLI di gioco nei bucket S3. Per supportare questo caso d'uso, puoi creare un ruolo IAM con accesso appropriato per leggere e scrivere contenuti di gioco nei tuoi bucket S3 e associarli alle EC2 istanze che ospitano il software e gli script.

È possibile definire politiche basate sulle risorse per il bucket e per oggetti specifici. Le policy dei bucket S3 sono associate a un bucket S3 e possono essere utilizzate per limitare l'accesso al bucket e agli oggetti al suo interno, nonché per concedere l'accesso alle risorse Amazon S3 da altri account. Ad esempio, in scenari in cui più team o diversi studi di sviluppo di giochi lavorano sugli stessi contenuti di gioco e richiedono lo stesso accesso ai contenuti ospitati centralmente in Amazon S3, puoi utilizzare una policy sui bucket S3 per definire le autorizzazioni per l'accesso tra account alle risorse S3. Prendi in considerazione l'utilizzo dei punti di accesso S3, che possono semplificare la gestione dell'accesso ai dati condivisi creando punti di accesso con nomi e autorizzazioni specifici per ogni applicazione o set di applicazioni. La documentazione di Amazon S3 contiene best practice aggiuntive per il controllo degli accessi in Amazon S3.

Granting short-term access to your content

Quando l'accesso è necessario solo per uno specifico periodo di tempo limitato, crea un programma temporaneo URLs che garantisca l'accesso a breve termine ai tuoi contenuti. Amazon S3 fornisce supporto per la generazione di oggetti predefiniti URLs, che consentono ai proprietari degli oggetti

di concedere un accesso limitato nel tempo agli oggetti in Amazon S3 senza aggiornare la policy sui bucket. In questo modo, l'utente finale o l'applicazione a cui viene concesso l'accesso non deve disporre di un account o di autorizzazioni IAM e utilizza invece l'URL predefinito per accedere al contenuto.

Si tratta di una procedura consigliata comunemente utilizzata in diversi casi d'uso dei giochi, ad esempio per concedere ai giocatori autorizzati l'accesso ai contenuti scaricabili a cui hanno diritto e fornire un accesso temporaneo ai contenuti di gioco per un periodo di tempo limitato. Presigned URLs può essere utilizzato anche per fornire autorizzazioni temporanee per il caricamento di contenuti in un bucket S3. Ad esempio, potresti prendere in considerazione l'utilizzo di un URL predefinito per fornire a un giocatore l'accesso per caricare i log dei client per aiutare il tuo team di supporto nella risoluzione di un caso di assistenza.

Utilizzare una rete di distribuzione dei contenuti per fornire l'accesso ai contenuti

Sebbene le applicazioni, gli sviluppatori di giochi, gli artisti e altro personale possano aver bisogno dell'accesso diretto ai contenuti dei bucket S3 per scopi di sviluppo e gestione, utilizza una rete di distribuzione dei contenuti per fornire l'accesso ai contenuti disponibili pubblicamente ai giocatori o ad altri utenti su Internet. Questo approccio migliora le prestazioni di download e riduce i costi memorizzando nella cache i contenuti a cui si accede di frequente. Amazon CloudFront può distribuire i tuoi contenuti a livello globale memorizzandoli nella cache e distribuendoli più vicino ai giocatori, riducendo al contempo il carico sull'origine di download del gioco, come Amazon S3.

Anziché distribuire i contenuti pubblici direttamente dai bucket S3, ti consigliamo di mantenerli privati e di pubblicarli pubblicamente utilizzando. CloudFront CloudFront può essere configurato per richiedere ai giocatori di accedere ai tuoi contenuti privati (ad esempio il download di un nuovo gioco solo per giocatori a pagamento) utilizzando cookie firmati URLs o firmati. Puoi quindi sviluppare l'applicazione per creare e distribuire cookie firmati URLs per utenti autenticati o per inviare intestazioni set-cookie che impostano cookie firmati per utenti autenticati. Quando crei cookie firmati URLs o firmati per controllare l'accesso ai tuoi file, puoi specificare una data e un'ora di fine, dopo le quali l'URL e i cookie non sono più validi.

Facoltativamente, puoi anche specificare l'indirizzo IP o l'intervallo di indirizzi dei computer che possono essere utilizzati per accedere ai tuoi contenuti, il che è utile se desideri limitare l'accesso a specifici partner di studi di sviluppo di giochi o reti di appaltatori. Utilizza i cookie firmati quando desideri fornire l'accesso a più file con restrizioni o se non desideri modificare quelli correnti. URLs Utilizza signed URLs quando desideri limitare l'accesso a singoli file o se gli utenti utilizzano un client che non supporta i cookie. I cookie firmati URLs hanno la precedenza sui cookie firmati.

Passaggi dell’implementazione

- Utilizza i ruoli IAM e le policy relative ai bucket per concedere l'accesso appropriato ai bucket S3 per applicazioni interne, team o scenari che coinvolgono più account.
- Genera impostazioni predefinite URLs per garantire l'accesso a breve termine agli oggetti S3, adatte per contenuti scaricabili o caricamenti temporanei come i log dei clienti.
- Usa Amazon CloudFront con cookie firmati URLs o firmati per fornire in modo più sicuro contenuti privati agli utenti autenticati

GAMESEC04-BP02 Limita l'accesso all'origine alle reti di distribuzione dei contenuti autorizzate () CDNs

Impedisci agli utenti di aggirare le tue reti di distribuzione dei contenuti per accedere direttamente ai contenuti dalla tua origine, come i bucket Amazon S3. È importante limitare l'accesso all'origine ai soli utenti autorizzati CDNs, in modo da ridurre i costi di trasferimento dei dati derivanti dalla distribuzione inutile di contenuti al di fuori dell'origine. Inoltre, migliora il livello di sicurezza mediante l'accesso pubblico ai contenuti di origine attraverso lo stesso punto di ingresso, dove è possibile implementare controlli di sicurezza all'avanguardia come il filtraggio di AWS WAF livello 7, l'inserimento e l'ispezione dei parametri di richiesta HTTP relativi alla sicurezza e le protezioni Distributed Denial of Service (S). DDo

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Per implementare questi controlli per un'origine Amazon S3, puoi utilizzare un'identità di accesso di CloudFront origine Amazon (OAI), che verifica che le richieste ai tuoi oggetti S3 provengano dalla tua distribuzione. CloudFront Associati alla tua CloudFront distribuzione per AWS WAF fornire filtri di livello 7. Tuttavia, se offrite contenuti aggiuntivi CDNs, potete configurare il CDN per inserire una o più intestazioni HTTP personalizzate nelle richieste di origine, che possono essere controllate AWS WAF per verificare che il traffico in entrata provenga dal vostro provider CDN autorizzato.

Questo approccio è utile anche per evitare che gli utenti eludano i provider CDN quando l'origine è ospitata su un Application Load Balancer (ALB). ALBs può essere associato a per protezioni di livello 7. AWS WAF Puoi configurare l'inserimento AWS WAF di un'intestazione HTTP personalizzata che verrà ispezionata dal tuo ALB per elaborare e ispezionare il traffico in entrata verso il sistema di bilanciamento del carico. AWS WAF

Esempio del cliente

AnyCompany Games implementa restrizioni di accesso all'origine per proteggere le risorse di gioco, i contenuti scaricabili e i file di patch da accessi diretti non autorizzati che potrebbero consentire ai giocatori di aggirare i controlli di sicurezza o ottenere contenuti premium senza un'autenticazione adeguata. Questo approccio consente loro di monitorare i modelli di accesso ai contenuti attraverso un punto centralizzato, semplificando l'identificazione di comportamenti di download sospetti che potrebbero indicare la presenza di attacchi coordinati o di ridistribuzione non autorizzata dei contenuti.

Passaggi dell’implementazione

- Usa Amazon CloudFront Origin Access Identity (OAI) per limitare l'accesso diretto agli oggetti S3
- AWS WAF Associati al CloudFront nostro ALB per fornire filtri di livello 7 e contribuire alla protezione dagli attacchi DDo S e dalle richieste dannose.
- Configura intestazioni HTTP personalizzate in Cloudfront per verificare che il traffico in entrata provenga da fonti autorizzate.

GAMESEC04-BP03 Implementa restrizioni geografiche per limitare l'accesso non autorizzato

Quando un giocatore richiede i tuoi contenuti, Amazon CloudFront fornisce i contenuti richiesti dalla edge location più vicina, indipendentemente da dove si trova il player. Tuttavia, in alcuni scenari potresti dover limitare il modo in cui i tuoi contenuti sono accessibili agli utenti in specifiche parti del mondo. Ad esempio, potresti avere una strategia di distribuzione dei giochi a rotazione che rilascia i contenuti in più fasi, oppure potresti dover rispettare i controlli di accesso specifici del paese. country-by-country

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Puoi utilizzare le restrizioni geografiche, note anche come blocchi geografici, per impedire ai giocatori di aree geografiche specifiche di accedere ai contenuti che stai distribuendo tramite una distribuzione. CloudFront Questa funzionalità consente di limitare l'accesso ai file associati a una distribuzione e di limitare l'accesso a livello di paese. In alternativa, è possibile utilizzare un servizio di geolocalizzazione di terze parti per limitare l'accesso a un sottoinsieme dei file associati a una distribuzione o per limitare l'accesso con una granularità più precisa rispetto al livello di paese.

Utilizzando le restrizioni CloudFront geografiche, puoi consentire ai tuoi giocatori di accedere ai tuoi contenuti solo se si trovano in uno dei paesi che si trovano in un elenco consentito di paesi approvati. Puoi anche impedire ai giocatori di accedere ai tuoi contenuti se si trovano in uno dei paesi che compaiono nell'elenco dei paesi vietati. Se viene ricevuta una richiesta da una località geografica bloccata, CloudFront restituirà al giocatore un codice di stato HTTP 403 Forbidden. È importante notare che questo metodo funziona bene per contenuti non sensibili e non deve essere utilizzato come protezione autonoma per informazioni personali o artefatti di gioco sensibili.

Passaggi dell'implementazione

- Utilizza le restrizioni CloudFront geografiche per consentire o negare l'accesso ai contenuti in base agli elenchi di autorizzazioni o negazioni a livello di paese.
- Restituisci un codice di stato HTTP 403 Forbidden per le richieste provenienti da aree geografiche bloccate.
- Evita di fare affidamento esclusivamente sulle restrizioni geografiche per proteggere contenuti sensibili o informazioni personali

GAMESEC04-BP04 Limita l'accesso ai contenuti con soluzioni di gestione dei diritti digitali (DRM)

Valuta la possibilità di limitare l'accesso ai contenuti di gioco utilizzando strumenti di crittografia avanzati come una soluzione di gestione dei diritti digitali (DRM). Questo tipo di soluzione può essere utilizzata per crittografare i contenuti privati e distribuire le chiavi di decrittografia ai giocatori autorizzati.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Le soluzioni DRM sono consigliate in situazioni in cui si desidera consentire ai giocatori di scaricare i contenuti di gioco in anticipo, ma non si desidera che possano accedervi o riprodurli fino a un orario prestabilito. Ad esempio, ciò è comune in situazioni in cui i giocatori possono preordinare un gioco e configurare il proprio client di gioco in modo che inizi automaticamente a scaricare i file crittografati in anticipo. Questa strategia verifica che il gioco sia scaricato e pronto per essere giocato una volta rilasciato ufficialmente. Dopo il rilascio del gioco, il client di gioco del giocatore può richiedere le chiavi di decrittografia alla soluzione di backend DRM in modo che possa decrittografare i file scaricati in precedenza e iniziare a giocare.

I sistemi DRM vengono utilizzati anche per bloccare la ridistribuzione e la manipolazione non autorizzate dei giochi dopo che sono stati scaricati e installati da un giocatore autorizzato. I sistemi DRM richiedono l'integrazione con l'origine per lo scambio delle chiavi di crittografia e l'autorizzazione dei giocatori a recuperare la chiave di decrittografia. I fornitori commerciali di DRM offrono una gamma di soluzioni con funzionalità e supporto per diversi dispositivi.

Passaggi dell'implementazione

- Utilizza le soluzioni DRM per crittografare i contenuti di gioco privati e distribuire le chiavi di decrittografia ai giocatori autorizzati.
- Abilita il download anticipato dei file crittografati per i giochi preordinati, sbloccando l'accesso con chiavi di decrittazione al momento del rilascio.
- Integra i sistemi DRM con l'origine per gestire le chiavi di crittografia e bloccare la ridistribuzione o la manipolazione non autorizzate dei contenuti.

Rilevamento

GAMESEC05: Come monitorate e analizzate il comportamento di utilizzo dei giocatori all'interno del gioco?

Il monitoraggio e l'analisi del comportamento di utilizzo dei giocatori sono essenziali per gli studi di gioco perché consentono di rilevare minacce alla sicurezza, imbrogli e altre forme di comportamento abusivo che potrebbero compromettere l'integrità del gioco e la sicurezza dei giocatori. Monitorando schemi come tassi di progressione insoliti, transazioni di gioco anomale o comportamenti di comunicazione sospetti, puoi identificare potenziali imbroglioni, account fraudolenti o minacce coordinate prima che abbiano un impatto significativo sull'esperienza del giocatore.

Best practice

- GAMESEC05-BP01 Implementa una strategia completa di raccolta dati per monitorare il comportamento dei giocatori
- GAMESEC05-BP02 Raccogli, archivia e analizza i registri di utilizzo dei giocatori per rilevare comportamenti inappropriati

GAMESEC05-BP01 Implementa una strategia completa di raccolta dati per monitorare il comportamento dei giocatori

Per mantenere un'esperienza positiva per i giocatori, implementa una strategia completa di raccolta e analisi dei dati. L'acquisizione, l'archiviazione e l'analisi dei dati pertinenti forniscono informazioni su come i giocatori interagiscono con le funzionalità del gioco e tra di loro. Questo approccio basato sui dati può guidare il processo decisionale, migliorare il coinvolgimento e la fidelizzazione dei giocatori, ottimizzare le strategie di monetizzazione e, in ultima analisi, migliorare l'esperienza complessiva del giocatore.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Implementa sistemi di raccolta dati per acquisire e registrare le azioni rilevanti dei giocatori, come sessioni di gioco, progressi, risultati, acquisti, interazioni con gli elementi di gioco e attività sociali. Raccogli dati lato server come il carico del server, il traffico di rete e i registri degli errori per monitorare le prestazioni tecniche e identificare potenziali problemi. Raccogli il feedback dei giocatori tramite sondaggi, forum, ticket di assistenza e canali di social media per comprendere le loro esperienze e preferenze.

Quando archivia i dati di gioco, crea un data warehouse o un data lake centralizzato per archiviare e organizzare i dati raccolti e implementa pipeline per la pulizia, la trasformazione e l'aggregazione dei dati per preparare i dati per un'analisi efficiente.

Dopo aver archiviato i dati, analizzali per ottenere informazioni come la fidelizzazione e il tasso di abbandono dei giocatori, le strategie di monetizzazione e l'utilizzo delle funzionalità tramite strumenti di visualizzazione dei dati.

Passaggi dell’implementazione

- Registra e registra le azioni dei giocatori, le metriche lato server e il feedback per monitorare le interazioni e le prestazioni tecniche.
- Usa un data warehouse centralizzato come Amazon Redshift o S3 data lake per archiviare, pulire, trasformare e organizzare i dati di gioco per l'analisi.
- Analizza i dati raccolti con strumenti di visualizzazione, come Amazon Quicksight, per ottenere informazioni sulla fidelizzazione dei giocatori, sulla monetizzazione e sull'utilizzo delle funzionalità.

GAMESEC05-BP02 Raccogli, archivia e analizza i registri di utilizzo dei giocatori per rilevare comportamenti inappropriati

Usa il tuo gioco per raccogliere registri per capire come i giocatori utilizzano le funzionalità del gioco e come interagiscono con gli altri giocatori. Puoi quindi bloccare le attività non autorizzate che possono peggiorare l'esperienza del giocatore.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Invia eventi di registro strutturati alla Game Analytics Pipeline, utilizzando una soluzione di registrazione come Amazon CloudWatch Logs o Amazon OpenSearch Service o tramite una soluzione di un AWS partner come Datadog, Sumo Logic, New Relic, Honeycomb.io o Splunk. Struttura questi registri di utilizzo dei giocatori in modo che possano essere utilizzati per rilevare quando è necessario indagare su azioni specifiche dei giocatori.

Dopo aver acquisito i dati, valuta la possibilità di implementare strumenti per rilevare comportamenti di utilizzo inappropriati. Ad esempio, se il gioco include funzionalità social come la messaggistica all'interno del gioco, la chat vocale o i forum online, salva i registri delle interazioni di questi giocatori in un formato che possa essere analizzato per scopi di moderazione.

Configura la funzione di chat vocale del gioco per esportare le registrazioni su Amazon S3 e usa Amazon Transcribe per convertire la voce audio in formato testo che può essere memorizzato per l'elaborazione. In alternativa, puoi eseguire la trascrizione in streaming in tempo reale integrando il servizio di chat vocale del backend di gioco direttamente con l'API Transcribe per trascrivere l'audio in streaming in tempo reale. I team di moderazione possono esaminare manualmente il contenuto e, una volta che il contenuto è in un formato standard, puoi anche utilizzare i servizi AI/ML per eseguire la moderazione automaticamente. AWS Amazon Comprehend può essere utilizzato per eseguire l'elaborazione del linguaggio naturale (NLP) per scoprire informazioni dal testo non strutturato, che può classificare e organizzare le conversazioni in argomenti pertinenti e identificare comportamenti inappropriati come parolacce.

Passaggi dell'implementazione

- Raccogli, archivia e analizza i registri di utilizzo dei giocatori.
- Utilizza AWS i servizi per l'intelligenza artificiale e l'apprendimento automatico per esaminare e ottenere informazioni in modo più efficiente sui registri di utilizzo dei giocatori.

Protezione dell’infrastruttura

Fai riferimento al white paper Well-Architected Framework per le migliori pratiche di protezione dell'infrastruttura per la sicurezza che si applicano ai carichi di lavoro dei giochi.

GAMESEC06: Come monitorate e rispondete alle minacce all'infrastruttura?

Il monitoraggio e la risposta alle minacce all'infrastruttura sono fondamentali per gli studi di gioco perché la loro infrastruttura rappresenta la spina dorsale che supporta milioni di giocatori simultanei, elabora transazioni con denaro reale e archivia preziosi dati sui giocatori e contenuti di gioco proprietari. Per gli studi di gioco è essenziale implementare sistemi di monitoraggio e processi di risposta agli incidenti in grado di preservare l'integrità sia delle esperienze dei giocatori che delle operazioni aziendali.

Best practice

- GAMESEC06-BP01 Usa strumenti per rilevare e rispondere alle minacce alla tua infrastruttura
- GAMESEC06-BP02 Usa strumenti di intelligenza artificiale e apprendimento automatico per automatizzare gli aspetti della tua strategia di protezione dell'infrastruttura
- GAMESEC06-BP03 Usa le informazioni dei log a livello di sistema per migliorare continuamente la tua strategia di protezione dell'infrastruttura

GAMESEC06-BP01 Usa strumenti per rilevare e rispondere alle minacce alla tua infrastruttura

Per monitorare continuamente le attività dannose e i comportamenti non autorizzati all'interno del tuo AWS ambiente, prendi in considerazione l'utilizzo di Amazon. GuardDuty GuardDuty identifica le minacce monitorando il comportamento degli account, l'attività di rete e i modelli di accesso ai dati all'interno del tuo ambiente. Analizza gli eventi su più fonti di dati, come log di CloudTrail eventi, Amazon VPC Flow Logs e DNS per potenziali minacce. Grazie all'integrazione con Amazon CloudWatch Events e Lambda GuardDuty , gli avvisi possono essere inoltrati automaticamente ai team di sicurezza competenti per ulteriori analisi.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

AWS Security Hub CSPMfornisce una visione completa dello stato di sicurezza AWS e verifica l'ambiente rispetto agli standard e alle best practice del settore della sicurezza. Security Hub CSPM raccoglie dati sulla sicurezza da tutti Account AWS i servizi e i prodotti partner di terze parti supportati, analizza le tendenze in materia di sicurezza e identifica i problemi di sicurezza con la massima priorità. L' GuardDutyintegrazione di Amazon con Security Hub CSPM ti consente di inviare i risultati da GuardDuty Security Hub CSPM. Security Hub CSPM può quindi includere tali risultati nell'analisi del livello di sicurezza dell'utente.

È normale che i malintenzionati utilizzino i bot per prendere il controllo degli account e imbrogliare nei giochi. WAF Bot Control ti offre visibilità e controllo sul traffico bot comune e pervasivo che può consumare risorse in eccesso, alterare le metriche, causare tempi di inattività o svolgere altre attività indesiderate.

Il ransomware è un codice dannoso progettato per ottenere l'accesso non autorizzato a sistemi e set di dati e crittografare tali dati per bloccare l'accesso da parte di giocatori legittimi. Dopo che il ransomware ha bloccato i giocatori dai loro sistemi e crittografato i loro dati sensibili, i criminali informatici chiedono un riscatto prima di fornire una chiave di decrittazione per sbloccare i dati. Organizations può essere completamente disattivata da un evento malevolo, con conseguenti costi significativi e perdita di produttività aziendale. Consulta la sezione Protezione Cloud AWS dell'ambiente dal ransomware per le migliori pratiche da applicare per rafforzare la capacità di combattere il ransomware prima, durante e dopo che si verifichi un incidente.

Il gioco può offrire ai giocatori la possibilità di contattare gli agenti dell'assistenza giocatori tramite un call center come Amazon Connect o chat bot che utilizzano Amazon Lex. Amazon Connect fornisce supporto per il monitoraggio delle conversazioni live e registrate. Per analizzare le interazioni tra i giocatori e i chatbot di supporto ai giocatori creati con Amazon Lex, puoi archiviare i log delle conversazioni di queste interazioni in Amazon CloudWatch Logs, che possono essere esportati in Amazon S3 e analizzati come descritto in precedenza.

Infine, esegui esercizi di penetration test come parte della tua strategia di protezione dell'infrastruttura. Sia che eseguiate queste valutazioni internamente o tramite un AWS partner, rispettate le politiche di assistenza AWS clienti per i test di penetrazione.

Passaggi dell’implementazione

- Usa Amazon GuardDuty per monitorare il comportamento degli account, l'attività di rete e i modelli di accesso ai dati alla ricerca di minacce e integra con Security Hub CSPM per una visione di sicurezza unificata.
- Implementa AWS WAF Bot Control per rilevare e mitigare il traffico di bot che può danneggiare le risorse e le esperienze dei giocatori.
- Conduci regolarmente esercizi di penetration test, aderendo alle politiche di assistenza AWS clienti, per valutare e rafforzare il tuo livello di sicurezza.

GAMESEC06-BP02 Usa strumenti di intelligenza artificiale e apprendimento automatico per automatizzare gli aspetti della tua strategia di protezione dell'infrastruttura

Amazon Lookout for Metrics utilizza l'apprendimento automatico per rilevare e diagnosticare automaticamente le anomalie nei dati aziendali e operativi e monitora le metriche più importanti per le tue attività con maggiore velocità e precisione. Il servizio semplifica anche la diagnosi della causa principale delle anomalie, come un calo improvviso delle entrate, degli accessi, delle transazioni o della fidelizzazione. Non richiede agli sviluppatori di giochi di avere esperienza di machine learning per la configurazione e può connettersi a fonti di dati popolari tra cui Amazon S3, Amazon, CloudWatch Amazon RDS, Amazon Redshift e molte applicazioni SaaS. Ad esempio, puoi integrare Amazon Lookout for Metrics con la Game Analytics Pipeline e altre fonti di dati per iniziare ad analizzare il comportamento per rilevare anomalie.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

In alternativa, puoi scegliere di creare, addestrare e ospitare un modello di machine learning personalizzato utilizzando l' SageMaker intelligenza artificiale di Amazon per affrontare casi d'uso come moderazione dei contenuti, rilevamento della tossicità, rilevamento di cheat, rilevamento di frodi e altro ancora.

Esempio del cliente

AnyCompany Games utilizza Amazon Lookout for Metrics per rilevare automaticamente modelli insoliti nelle prestazioni del server, nei tentativi di accesso dei giocatori o nei volumi di transazioni che

potrebbero indicare minacce da parte di malintenzionati. Inoltre, hanno utilizzato Amazon SageMaker AI per sviluppare modelli di machine learning personalizzati che analizzano continuamente i modelli di traffico di rete e il comportamento dei giocatori per aiutare a identificare minacce coordinate, come le reti di bot che tentano di sfruttare la loro economia virtuale.

Questo approccio automatizzato consente al team di sicurezza di concentrarsi sull'indagine e sulla risposta alle minacce autentiche anziché sul monitoraggio manuale di migliaia di parametri, assicurandosi al contempo che i modelli di minaccia emergenti vengano rilevati e risolti prima che possano influire in modo significativo sulla disponibilità del gioco o sulla sicurezza dei giocatori.

Passaggi dell’implementazione

- Usa Amazon Lookout for Metrics per rilevare e diagnosticare automaticamente le anomalie nei dati aziendali e operativi chiave
- Integra Amazon Lookout for Metrics con fonti di dati come Game Analytics Pipeline, Amazon S3 o per monitorare parametri come CloudWatch entrate, accessi e fidelizzazione.
- Usa Amazon SageMaker AI per creare, addestrare e ospitare modelli di machine learning personalizzati per casi d'uso avanzati come il rilevamento di cheat, la prevenzione delle frodi e la moderazione dei contenuti.

GAMESEC06-BP03 Usa le informazioni dei log a livello di sistema per migliorare continuamente la tua strategia di protezione dell'infrastruttura

Acquisisci e archivia i log a livello di sistema dai servizi pertinenti, come i registri di accesso ai server S3, i registri di accesso e i registri di CloudFrontaccesso ALB. Questi log possono essere archiviati in un bucket S3 nel tuo account e sono utili per associare le informazioni sull'utilizzo del giocatore dall'interno del gioco a informazioni a livello di sistema, tra cui dettagli di connessione come indirizzi IP, intestazioni di richiesta e relative manipolazioni e filtri delle richieste che potresti aver configurato nel backend di gioco. Puoi inviare questi log alle stesse soluzioni di registrazione menzionate in precedenza e analizzarli utilizzando query SQL con Amazon Athena senza richiedere lo spostamento dei log da Amazon S3.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Access Analyzer for S3 è una funzionalità che monitora le policy di accesso ai bucket, assicurandosi che le policy forniscano solo l'accesso previsto alle risorse Amazon S3. Access Analyzer for S3

valuta le politiche di accesso ai bucket e consente di individuare e correggere rapidamente i bucket con accessi potenzialmente non intenzionali.

Passaggi dell'implementazione

- Utilizza AWS i servizi per il rilevamento delle minacce e la risposta agli incidenti per automatizzare alcuni aspetti della strategia di protezione dell'infrastruttura.
- Ottieni informazioni sulla protezione dell'infrastruttura tramite log e AWS servizi a livello di sistema per l'intelligenza artificiale e l'apprendimento automatico.

Protezione dei dati

Quando sviluppi e progetti il tuo gioco, considera il tipo di dati che il tuo studio sta raccogliendo e come hai deciso di proteggerli. Gli argomenti da approfondire in merito a questo aspetto della sicurezza includono:

- In che modo hai scelto di identificare e classificare i tuoi dati
- Come state proteggendo i dati inattivi
- Come proteggete i dati in transito

Non esistono buone pratiche di protezione dei dati specifiche per Games Lens. Consulta il white paper Well-Architected Framework per le migliori pratiche di protezione dei dati per la sicurezza.

Risposta agli incidenti

GAMESEC07: Come state definendo e applicando le politiche per rispondere alla cattiva condotta e al comportamento abusivo dei giocatori?

La cattiva condotta e i comportamenti offensivi dei giocatori possono influire in modo significativo sull'esperienza dei giocatori. Il comportamento scorretto dei giocatori può allontanare i giocatori legittimi, con conseguente riduzione della fidelizzazione dei giocatori, riduzione delle entrate derivanti dagli acquisti in-game e recensioni negative che possono danneggiare la reputazione del gioco e le vendite future.

Definisci politiche che promuovano azioni positive tra i tuoi giocatori e stabilisci come applicarle.

Best practice

- GAMESEC07-BP01 Implementa un piano di risposta agli incidenti per gestire i cattivi attori e i comportamenti abusivi
- GAMESEC07-BP02 Blocca gli account associati a malintenzionati

GAMESEC07-BP01 Implementa un piano di risposta agli incidenti per gestire i cattivi attori e i comportamenti abusivi

Crea un piano d'azione per rispondere ai cattivi attori e ai comportamenti offensivi presenti nel gioco.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all’implementazione

Prendi in considerazione fattori come quando sospendere temporaneamente o bannare definitivamente i giocatori e per quanto tempo disabilitare le credenziali per i giocatori temporaneamente sospesi.

Esempio del cliente

AnyCompany Games crea un sistema di risposta agli incidenti a più livelli in cui infrazioni minori, come messaggi di chat inappropriati, comportano la sospensione automatica dell'account 24 ore su 24, mentre le violazioni più gravi come imbrogli o molestie innescano sospensioni immediate di 7 giorni con revisione obbligatoria da parte di moderatori umani.

Inoltre, AnyCompany Games stabilisce procedure di escalation in base alle quali i recidivi sono soggetti a sospensioni progressivamente più lunghe. Creano procedure di ricorso che consentono ai giocatori falsamente segnalati di contestare azioni automatizzate, mantenendo al contempo la sicurezza grazie ai requisiti di verifica dell'identità.

GAMESEC07-BP02 Blocca gli account associati a malintenzionati

Se lasciati inalterati, i comportamenti offensivi in un gioco possono continuare ad avere un impatto negativo sull'esperienza di gioco degli altri e devono essere mitigati il prima possibile. Implementa una procedura per imporre divieti o altre forme di restrizioni ai malintenzionati che hanno confermato di aver violato i tuoi termini di servizio.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

In genere, le regole e il processo di valutazione per determinare le circostanze per l'imposizione di questo tipo di restrizioni saranno determinati da personale, ad esempio un team della community di giocatori o un team di fiducia e sicurezza all'interno dell'organizzazione. Dopo aver segnalato i cattivi attori, esegui un flusso di lavoro predeterminato per agire sui giocatori identificati.

Ad esempio, AWS Step Functionsle AWS Lambdafunzioni possono essere utilizzate per eseguire un flusso di lavoro automatizzato che accetta un batch di account di giocatori come input. Il flusso di lavoro aggiorna quindi le voci in una tabella Amazon DynamoDB denominata Bans, che può includere dettagli sull'account del giocatore, il motivo del ban e la durata.

A seconda del design del tuo sistema di gestione del gioco e degli account e del tipo di abuso che subisci da parte di malintenzionati, mantieni un sistema di registrazione dei ban separato dal sistema di gestione degli account. Potresti non voler disattivare l'account del giocatore dal tuo sistema di gestione dell'account, preferendo invece semplicemente disattivare la sua capacità di giocare al tuo gioco. Ciò può essere utile in situazioni in cui le credenziali dell'account del giocatore vengono utilizzate per accedere a più giochi con termini di servizio o politiche diversi.

Passaggi dell’implementazione

- Definisci e applica le politiche per rispondere ai comportamenti abusivi da parte di malintenzionati.
- Utilizza AWS i servizi per automatizzare le tue risposte ai malintenzionati.

Resources

- AWS Guida tecnica sulla risposta agli incidenti di sicurezza
- AWS Blog sul Machine Learning: individua utenti reali e reali e scoraggia i malintenzionati con Amazon Rekognition Face Liveness
- AWS Soluzioni per i giochi: Community Health

Sicurezza delle applicazioni

GAMESEC08: Come proteggete la vostra pipeline? CI/CD

Una CI/CD pipeline di sviluppo di giochi è in genere composta da server e storage per il controllo del codice sorgente ad alta disponibilità, risorse di calcolo per eseguire le build e software per eseguire test automatici, oltre alla corretta connettività di rete delle macchine di sviluppo. Proteggere la CI/CD pipeline è importante per proteggere le informazioni sensibili, preservare l'integrità del codice e mantenere versioni affidabili. L'integrazione di governance e guardrail consente l'agilità degli sviluppatori mantenendo al contempo buone pratiche di sicurezza.

Poiché i giochi spesso gestiscono l'elaborazione dei pagamenti, archiviano informazioni personali e mantengono economie virtuali che valgono soldi veri, una violazione della sicurezza nel processo di sviluppo potrebbe comportare perdite finanziarie significative, sanzioni normative e una perdita di fiducia dei giocatori.

Integrando le misure di sicurezza, le organizzazioni mantengono la visibilità e il controllo sul processo di distribuzione del software, consentendo una risposta rapida agli incidenti e promuovendo una cultura di pratiche di codifica sicure.

Best practice

- GAMESEC08-BP01 Applica la sicurezza in ogni fase della pipeline CI/CD

GAMESEC08-BP01 Applica la sicurezza in ogni fase della pipeline CI/CD

Le barriere, come i controlli degli accessi, la separazione delle mansioni e gli audit trail, forniscono protezione da accessi non autorizzati o attività dannose.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Anche il personale, i processi e la tecnologia devono proteggere la pipeline. Le persone più vicine al codice devono stabilire pratiche di codifica sicure e assicurarsi che le seguano. Eseguite continuamente i vostri processi per verificare che vi sia coerenza nel livello di sicurezza in tutta la pipeline. Infine, implementa la tecnologia per verificare che le migliori pratiche e i processi non vengano aggirati.

Esempio del cliente

AnyCompany Games implementa controlli di accesso basati sui ruoli, in base ai quali solo gli sviluppatori senior possono approvare le modifiche al codice del sistema anti-cheat, mentre richiede ai membri del team di sicurezza una revisione obbligatoria del codice per i componenti che gestiscono i dati di pagamento dei giocatori.

La loro CI/CD pipeline esegue automaticamente controlli di convalida dei modelli di minaccia, assicurandosi che nuove funzionalità, come un marketplace per la negoziazione dei giocatori, vengano testate rispetto a vettori di attacco precedentemente identificati, come exploit di duplicazione di oggetti o tentativi di transazione fraudolenta.

Passaggi dell'implementazione

- Fornisci le autorizzazioni agli utenti in base al principio del privilegio minimo.
- AWS CloudTrail Da utilizzare per controllare le chiamate API effettuate tra i servizi utilizzati nella pipeline.
- Utilizza gli hook di pre-commit per verificare che il codice segua le pratiche generali e le politiche aziendali.

Automatizza la sicurezza

GAMESEC09: Come automatizzate la sicurezza all'interno della vostra pipeline? CI/CD

Integra le misure di sicurezza all'interno della tua CI/CD pipeline per mantenere una solida posizione di sicurezza durante tutto il ciclo di vita dello sviluppo. Questo processo offre molti degli stessi vantaggi della sicurezza della pipeline. Avere una CI/CD pipeline sicura riduce la probabilità che si verifichino eventi di sicurezza che potrebbero potenzialmente ritardare la tempistica di sviluppo del gioco.

Garantire la sicurezza nella CI/CD pipeline implica l'implementazione delle migliori pratiche e strumenti di sicurezza in ogni fase del ciclo di sviluppo. L'implementazione della sicurezza consente inoltre di ridurre i tempi di revisione della sicurezza.

L'automazione della sicurezza all'interno della CI/CD pipeline è particolarmente importante per verificare che i controlli di sicurezza siano implementati e testati in modo coerente ad ogni modifica del codice. L'implementazione degli strumenti e dell'automazione adeguati può offrire giochi sicuri e protetti.

Best practice

- GAMESEC09-BP01 Integra strumenti e automazione per ridurre il tempo medio delle revisioni di sicurezza

GAMESEC09-BP01 Integra strumenti e automazione per ridurre il tempo medio delle revisioni di sicurezza

Per identificare le vulnerabilità di sicurezza, le organizzazioni possono utilizzare una varietà di strumenti e servizi diversi come Static Application Security Testing (SAST) e Dynamic Application Security Testing (DAST). SAST è un modo per esaminare il codice sorgente e determinare le vulnerabilità di sicurezza. DAST è un metodo a scatola nera per testare il codice che verifica le applicazioni senza guardare il codice sorgente.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Un altro strumento che le organizzazioni possono utilizzare è l'analisi della composizione del software (SCA), che valuta la sicurezza delle dipendenze da terze parti o open source. Per un approccio più manuale, è possibile implementare revisioni sicure del codice in tutta la pipeline.

Esempio del cliente

AnyCompany Games utilizza gli strumenti SAST per segnalare automaticamente potenziali falle di sicurezza durante il processo di sviluppo. Inoltre, utilizzano gli strumenti DAST per simulare le minacce contro l'esecuzione di build di giochi per verificare che i controlli di sicurezza funzionino come previsto. Inoltre, AnyCompany Games integra strumenti di scansione delle dipendenze nel proprio processo di sviluppo per identificare automaticamente le vulnerabilità note nelle librerie e nei motori di gioco di terze parti.

Passaggi dell'implementazione

- Usa Amazon CodeGuru come strumento SAST.
- Usa strumenti open source come OWASP Dependency Check o. SonarQube OWASPZap

Resources

- Sicurezza per gli sviluppatori

Modellazione delle minacce

GAMESEC10: Come si integra la modellazione delle minacce nel ciclo di vita di sviluppo delle applicazioni della propria organizzazione?

La modellazione delle minacce è il processo di identificazione e definizione delle priorità delle potenziali minacce alle applicazioni e di determinazione delle soluzioni che possono essere utilizzate per mitigarle. Questa pratica è diventata sempre più importante man mano che i giochi si sono evoluti in sistemi complessi e connessi che gestiscono dati sensibili degli utenti e transazioni con denaro reale.

Integra la modellazione delle minacce come esercizio continuo per supportare la sicurezza del gioco, non solo nella fase iniziale di progettazione, ma man mano che il gioco continua a crescere ed evolversi.

Best practice

- GAMESEC10-BP01 Determina quando e come completare gli esercizi di modellazione delle minacce durante tutto il ciclo di vita dello sviluppo delle applicazioni

GAMESEC10-BP01 Determina quando e come completare gli esercizi di modellazione delle minacce durante tutto il ciclo di vita dello sviluppo delle applicazioni

Non esiste un unico modo migliore per affrontare la modellazione delle minacce. I dettagli su quando e come eseguire questa operazione variano in base alle esigenze specifiche del tuo studio di gioco. Ad esempio, a seconda delle dimensioni del tuo studio, potresti avere membri del team coinvolti in uno o più aspetti del processo di modellazione delle minacce.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Il AWS Security Blog offre una panoramica delle considerazioni da tenere a mente quando si elabora una strategia per la modellazione delle minacce, ad esempio:

- Quali membri del team e quali personaggi dovrebbero essere coinvolti nella modellazione delle minacce
- Come determinare gli strumenti di flusso di lavoro appropriati da utilizzare
- Come determinare la titolarità dei vari aspetti della modellazione delle minacce
- Come identificare e valutare i controlli di sicurezza da utilizzare nella progettazione del carico di lavoro

Esempio del cliente

AnyCompany Games inizia catalogando risorse preziose come i dati dei giocatori, il codice e gli algoritmi di gioco, le valute di gioco, i contenuti generati dagli utenti e la proprietà intellettuale come contenuti inediti o motori proprietari. Considerano diversi tipi di potenziali malintenzionati, come gli imbroglioni che cercano vantaggi sleali, i malintenzionati che cercano di rubare dati personali o finanziari e gli utenti malintenzionati che cercano di interrompere il gioco.

Durante tutto il processo di sviluppo, AnyCompany Games utilizza modelli di minaccia per guidare pratiche di codifica sicure e influenzare le strategie di test per concentrarsi sulle aree ad alto rischio. Prima del lancio del gioco, eseguono revisioni complete sulla modellazione delle minacce per valutare la preparazione ai carichi di giocatori previsti e ai tentativi di accesso non autorizzati e per preparare le procedure di risposta agli incidenti.

Passaggi dell'implementazione

- Implementa i guardrail in ogni fase della pipeline. CI/CD
- Utilizza l'automazione e gli strumenti per migliorare l'efficienza delle revisioni della sicurezza delle applicazioni.
- Utilizzate la modellazione delle minacce come processo per migliorare la sicurezza delle vostre applicazioni.

Resources

- AWS Blog sulla sicurezza: Come affrontare la modellazione delle minacce
- NIST: Guida alla modellazione delle minacce di sistema incentrata sui dati
- Modellazione delle minacce nel modo giusto per i costruttori: formazione autonoma virtuale di Skill Builder AWS
- Modellazione delle minacce per i costruttori — Workshop AWS

Resources

Consulta le seguenti risorse per saperne di più sulle nostre best practice relative alla sicurezza.

Documenti correlati:

- Scenari comuni di Amazon Cognito
- Utilizzo di signed URLs
- Usa i flussi di canale per rimuovere volgarità e contenuti sensibili dai messaggi nella messaggistica SDK di Amazon Chime
- Sicurezza in Amazon GameLift
- Distribuzione sicura dei contenuti con Amazon CloudFront
- Guida alla risposta alla sicurezza
- AWS Le migliori pratiche per la resilienza DDo S
- Proteggere l' Cloud AWS ambiente dal ransomware

Soluzioni partner correlate:

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

Materiali di formazione correlati:

- Guida introduttiva ad Amazon Cognito
- Formazione autonoma sulla sicurezza

Affidabilità

Il pilastro dell'affidabilità include la capacità di un sistema di riprendersi da interruzioni dell'infrastruttura o del servizio, acquisire dinamicamente risorse di elaborazione per soddisfare la domanda e mitigare interruzioni come configurazioni errate o problemi transitori di rete.

Aree di interesse

- Principi di progettazione
- Fondamenti
- Architettura del carico di lavoro
- Gestione delle modifiche
- Gestione dei guasti
- Resources

Principi di progettazione

Oltre ai principi di progettazione contenuti nel white paper AWS Well-Architected Framework, i seguenti sono principi di progettazione che possono aumentare l'affidabilità nel cloud per i carichi di lavoro dei giochi:

- Stabilisci la linea di base per gli obiettivi di massima concorrenza tra giocatori e scalabilità del sistema necessari per soddisfare le proiezioni aziendali: prima del lancio di un gioco e durante le operazioni di gioco dal vivo, sviluppa stime del numero di giocatori simultanei previsto durante il picco per stabilire gli obiettivi di scalabilità del sistema necessari per soddisfare queste proiezioni. Questo aiuta a creare una base per l'affidabilità del gioco. Definisci politiche di scalabilità per soddisfare automaticamente le variazioni della domanda senza influire sulla disponibilità verificando che i tuoi sistemi di scalabilità gestiscano correttamente le sessioni attive dei giocatori.
- Misura la tua affidabilità e l'impatto sull'esperienza del giocatore: definisci gli indicatori chiave di prestazione (KPIs) che rappresentino lo stato di salute del tuo gioco. Monitora l'impatto dei cambiamenti nell'infrastruttura e nelle funzionalità di gioco sulla tua affidabilità.

Fondamenti

Per raggiungere l'affidabilità, un sistema deve avere una base ben pianificata e un monitoraggio con meccanismi per la gestione delle variazioni della domanda o dei requisiti. Il sistema deve essere progettato in modo da rilevare guasti e ripararsi automaticamente.

Non esistono best practice di base specifiche per Games Lens. Fai riferimento al white paper di Well-Architected Framework per le migliori pratiche relative ai fondamenti dell'affidabilità applicabili ai carichi di lavoro dei giochi.

Architettura del carico di lavoro

GAMEREL01: La tua architettura di gioco sfrutta la resilienza del cloud?

AWS l'infrastruttura è costruita attorno a regioni e zone di disponibilità. Regioni AWS forniscono più zone di disponibilità fisicamente separate e isolate, collegate tramite reti a bassa latenza, ad alto throughput e altamente ridondanti. Questi costrutti possono essere utilizzati per progettare carichi di lavoro incentrati sugli obiettivi di affidabilità.

Best practice

- GAMEREL01-BP01 Distribuisci l'infrastruttura di gioco su più zone e regioni di disponibilità per migliorare la resilienza

GAMEREL01-BP01 Distribuisci l'infrastruttura di gioco su più zone e regioni di disponibilità per migliorare la resilienza

Per ridurre al minimo l'impatto dei problemi dell'infrastruttura localizzata sui giocatori, è necessario distribuire la distribuzione dell'infrastruttura in modo uniforme su un numero sufficiente di postazioni indipendenti da poter resistere a problemi imprevisti pur avendo una capacità sufficiente per soddisfare le esigenze della domanda dei giocatori.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Quando si implementa l'infrastruttura di gioco, si consiglia di distribuire uniformemente la capacità tra più zone di disponibilità di una regione in modo da poter resistere alle interruzioni di una o più zone di disponibilità senza interrompere l'esperienza del giocatore. I servizi di backend di gioco, come le applicazioni Web, devono essere bilanciati in base al carico su più zone di disponibilità o devono essere creati utilizzando servizi gestiti come AWS Lambda Amazon API Gateway, che forniscono un'alta disponibilità regionale fin dalla progettazione. Allo stesso modo, i componenti che mantengono lo stato come cache, database, code di messaggi e soluzioni di storage devono essere progettati per fornire una persistenza duratura dei dati su più zone di disponibilità, che è fornita dalla progettazione in servizi come Amazon S3, DynamoDB e Amazon SQS e può essere configurata in altri servizi.

Quando progetti l'architettura di hosting dei server di gioco per garantire la resilienza, distribuisci le tue flotte di server di gioco in modo uniforme tra le zone di disponibilità all'interno di e per massimizzare l'accesso alla capacità di elaborazione disponibile nella regione e ridurre l'ambito di impatto delle limitazioni della zona di disponibilità. Regione AWS Ad esempio, puoi configurare Amazon EC2 Auto Scaling per utilizzare le zone di disponibilità. Se un' EC2istanza non funziona correttamente, EC2 Auto Scaling può sostituirla e avviare le istanze in altre zone di disponibilità se una o più zone di disponibilità non sono disponibili.

Per infrastrutture critiche, come l'autenticazione, fornisci un numero minimo di istanze valide in esecuzione su più zone di disponibilità e utilizza la scalabilità automatica per gestire gli aumenti di carico o la tolleranza agli errori in caso di problemi con una delle zone di disponibilità.

Implementa la tua infrastruttura di gioco in più regioni per massimizzare la disponibilità. Le funzionalità di disaster recovery interregionali come i database globali Aurora e l'infrastruttura ridondante che può diventare attiva con una semplice modifica del DNS implementata in una regione secondaria possono garantire la continuità del servizio in caso di compromissione della regione primaria. Sebbene incoraggiamo questa opzione per consentire ai servizi di backend di gioco di raggiungere un'elevata disponibilità, questa raccomandazione è particolarmente importante per i server di gioco.

Ad esempio, in una partita multiplayer, è probabile che la capacità dell'infrastruttura per i server di gioco superi quella necessaria per gli altri servizi, poiché i server di gioco vengono utilizzati per ospitare sessioni di gioco per i giocatori. Molti giochi scelgono di suddividere i giocatori in regioni di gioco logiche (come gli Stati Uniti occidentali e orientali). Per semplificare l'esperienza del giocatore e semplificare l'utilizzo dell'infrastruttura globale per ospitare i giochi, prendi in considerazione la possibilità di disaccoppiare il nome delle aree di gioco rivolte ai giocatori dalla regione del provider

cloud sottostante o dalla sede del data center che ospita fisicamente i server di gioco, insieme ad altre infrastrutture come Local Zones o i tuoi data center che ospitano istanze di server di gioco che supportano quella regione di gioco.

Quando progetti il tuo servizio di matchmaking, implementa un'architettura multiregionale con distribuzioni software separate tra le regioni. Separa l'implementazione del tuo servizio di matchmaking dalle flotte che ospitano le istanze dei tuoi server di gioco in modo da poter indirizzare i giocatori a un server di gioco nelle Regioni indipendentemente dalla distribuzione regionale del tuo servizio di matchmaking che ha gestito la richiesta di matchmaking.

Progetta una logica nell'implementazione del matchmaking per favorire le regioni dei server di gioco che soddisfano la tua latenza e altre regole, con la possibilità di reindirizzare i giocatori verso altre regioni se la capacità delle tue flotte è scarsa o ci sono altre interruzioni dell'infrastruttura regionale.

Passaggi dell’implementazione

- Distribuisci l'infrastruttura di gioco in modo uniforme su più zone di disponibilità per garantire disponibilità e resilienza elevate.
- Implementa servizi di backend di gioco e componenti stateful utilizzando sistemi gestiti come AWS Lambda Amazon S3, DynamoDB e SQS, oppure configura il bilanciamento del carico e la durabilità per soluzioni personalizzate.
- Implementa implementazioni multiregionali per server e servizi di gioco critici, utilizzando soluzioni di disaster recovery come i database globali Aurora e le regioni logiche rivolte ai giocatori disaccoppiate dalle posizioni fisiche sottostanti.

Resources

- Stabilità statica
- Le migliori pratiche per le code delle sessioni di GameLift gioco di Amazon
- Flotte Amazon GameLift multiregionali
- Database globale Aurora per il disaster recovery

Gestione delle modifiche

GAMEREL02: Come si fa a scalare i giochi stateful per soddisfare i cambiamenti della domanda?

Poiché la domanda dei giocatori varia nel tempo, la tua infrastruttura di gioco dovrebbe essere in grado di adattarsi in modo adattivo per gestire questi requisiti in continua evoluzione. Sebbene sia difficile prevedere in anticipo la popolarità di un gioco, è necessario progettare un approccio architettonico che consenta di aggiungere e rimuovere la capacità dell'infrastruttura per far fronte alle fluttuazioni della popolazione di giocatori.

Best practice

- GAMEREL02-BP01 Implementa una strategia di scalabilità che incorpori lo stato delle sessioni di gioco attive dei giocatori
- GAMEREL02-BP02 Supporta l'uso di più tipi di istanze per il tuo gioco EC2

GAMEREL02-BP01 Implementa una strategia di scalabilità che incorpori lo stato delle sessioni di gioco attive dei giocatori

Implementa una soluzione per scalare automaticamente la tua infrastruttura di gioco in modo da incorporare la natura statica delle sessioni con i giocatori in connessione attiva e gestire con eleganza le attività di scalabilità senza interrompere il gameplay.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Uno dei vantaggi dello sviluppo di un gioco nel cloud è l'elasticità che può essere ottenuta scalando automaticamente l'infrastruttura dei server in base alle esigenze per soddisfare la domanda. Sebbene i giochi e i servizi di backend stateless o asincroni possano essere EC2 scalati dinamicamente utilizzando le policy di Amazon Auto Scaling, l'autoscaling EKS o tecniche simili tipicamente adottate per le applicazioni web scalabili, gli sviluppatori di giochi in genere richiedono un approccio più personalizzato per scalare i giochi a stato o sincroni per aiutare a bloccare le interruzioni delle sessioni attive dei giocatori.

Passaggi dell'implementazione

- Per i giochi stateful, genera metriche personalizzate che possono essere utilizzate per monitorare lo stato delle sessioni di gioco e la capacità disponibile del server di gioco, che possono essere segnalate ad Amazon CloudWatch come metriche personalizzate. Esercitati con funzionalità di monitoraggio delle applicazioni, come CloudWatch Synthetics, e verifica che il gioco non comprometta funzioni che il semplice monitoraggio dello stato attivo e inattivo potrebbe non rilevare.

- Utilizzando metriche personalizzate, implementa un software di ridimensionamento del server di gioco, ad esempio, come applicazione serverless che utilizza AWS Lambda function oppure AWS Fargate, per gestire la flotta di istanze di server di gioco dedicate utilizzando l' AWS SDK per effettuare chiamate API per aggiornare le impostazioni di capacità minima, massima e desiderata per i gruppi di Auto Scaling EC2 che ospitano la build del tuo server di gioco.
- Usa Amazon GameLift per ospitare i tuoi server di gioco e utilizza le funzionalità di ridimensionamento automatico dei server di out-of-the-box gioco per gestire questo processo di scalabilità per te.

Le funzionalità di scalabilità automatica di Amazon GameLift riconoscono le sessioni attive dei giocatori e possono essere configurate per bloccare la chiusura o la scalabilità delle istanze di server di gioco che ospitano attivamente giocatori. Per ulteriori informazioni, consulta Monitorare GameLift i server Amazon con Amazon CloudWatch.

GAMEREL02-BP02 Supporta l'uso di più tipi di istanze per il tuo gioco EC2

Quando offri un gioco utilizzando EC2 istanze o se utilizzi contenitori ospitati su EC2 istanze del tuo sito Account AWS, utilizza più tipi di istanze nella tua strategia di hosting. Utilizzando più tipi di istanze, puoi aumentare il numero di opzioni di calcolo che possono essere utilizzate quando il gioco è in fase di scalabilità per aggiungere altri server per supportare la crescita dei giocatori, il che migliora l'affidabilità nel caso in cui il tipo di istanza preferito non sia disponibile.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Quando offri un gioco utilizzando istanze Spot, utilizza più tipi di istanze, poiché la disponibilità delle istanze Spot varia in base alla domanda dei clienti.

Testa il tuo gioco su più tipi di istanze per soddisfare i tuoi requisiti in termini di costi e prestazioni e stabilire una classifica prioritaria dei tipi di istanza. Amazon EC2 Auto Scaling supporta l'utilizzo di più tipi e dimensioni di istanze, nonché l'assegnazione di pesi a ciascun tipo di istanza nella configurazione, in modo da poter implementare una classificazione prioritaria delle opzioni di calcolo.

Quando offri un gioco utilizzando l'hosting GameLift gestito di Amazon, decidi di che tipo di istanze ti servono e come eseguire i processi del server di gioco su di esse (utilizzando una configurazione di runtime). Quando scegli le risorse per una flotta, considera diversi fattori, tra cui il sistema operativo di gioco, il tipo di istanza (l'hardware di elaborazione) e se utilizzare istanze On-Demand, istanze

Spot o entrambe. I costi di hosting con Amazon dipendono GameLift principalmente dal tipo di istanze utilizzate. Per ulteriori informazioni, consulta Scegliere le risorse di calcolo per una flotta gestita.

Passaggi dell'implementazione

- Usa più tipi di EC2 istanze per migliorare l'affidabilità e le opzioni di scalabilità quando ospiti il gioco su EC2 o container.
- Configura Amazon EC2 Auto Scaling o GameLift flotte con tipi e pesi di istanze con priorità per ottimizzare costi e prestazioni.
- Testa il tuo gioco su vari tipi di istanze per verificare che le prestazioni soddisfino i requisiti e adatta di conseguenza la tua strategia di hosting.

Gestione dei guasti

GAMEREL03: Come si fa a mantenere lo stato del gioco durante le interruzioni dell'infrastruttura?

Poiché l'infrastruttura di gioco subisce vari eventi operativi nel tempo, l'architettura del gioco deve essere progettata per mantenere la continuità delle esperienze dei giocatori e preservare lo stato del gioco durante gli eventi dell'infrastruttura. Per gestire questi eventi, implementa meccanismi di monitoraggio, arresti automatici e meccanismi di persistenza dello stato per garantire un'esperienza di gioco fluida per i tuoi giocatori.

Best practice

- GAMEREL03-BP01 Monitora le interruzioni dei server di gioco e utilizza i dati per migliorare l'architettura di hosting per raggiungere gli obiettivi di affidabilità
- GAMEREL03-BP02 Implementa un accoppiamento libero delle funzionalità di gioco per gestire i fallimenti con un impatto minimo sull'esperienza del giocatore
- GAMEREL03-BP03 Monitora gli eventi dell'infrastruttura nel tempo per misurare l'impatto sul comportamento dei giocatori

GAMEREL03-BP01 Monitora le interruzioni dei server di gioco e utilizza i dati per migliorare l'architettura di hosting per raggiungere gli obiettivi di affidabilità

Monitora le metriche dei server di gioco e l'impatto di guasti o riduzioni delle prestazioni, come l'aumento della latenza sotto carico, sul comportamento dei giocatori nel tempo, in modo da poter adattare la strategia di hosting dei server di gioco per soddisfare i requisiti di affidabilità del gioco. L'infrastruttura del server di gioco da danneggiare deve essere rimossa immediatamente dal servizio se ciò ha un impatto sui giocatori o sostituita in modo proattivo quando non ci sono sessioni attive dei giocatori ospitate sul server.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Negli scenari in cui i giochi sono ospitati come REST APIs, l'affidabilità del sistema può essere gestita come le tradizionali architetture di applicazioni web, in cui il traffico può essere bilanciato del carico su più server in modo distribuito per mitigare il rischio di guasti del server.

Per un gameplay sincrono in tempo reale, una sessione di gioco è generalmente ospitata su un processo del server di gioco in esecuzione su una macchina virtuale o su un'istanza del server di gioco, poiché lo stato di gioco deve essere mantenuto in modo efficiente e replicato sui client di gioco connessi. Questa implementazione significa che l'esperienza del giocatore è strettamente legata alle prestazioni e all'affidabilità del processo del server di gioco che ospita la sessione di gioco. Questo tipo di architettura rende la gestione dell'affidabilità dei server di gioco più complessa rispetto agli approcci tradizionali.

Per mitigare l'impatto di un errore del server di gioco, configura il gioco in modo che esegua continuamente aggiornamenti asincroni dello stato di gioco di un giocatore su una cache o un database ad alta disponibilità come Amazon (ElastiCache Redis OSS) o Amazon MemoryDB. Se si verifica un errore del server, l'ultimo stato di gioco salvato del giocatore può essere recuperato dall'archivio dati esterno e la sessione può essere ripristinata su una nuova istanza del server di gioco.

Tuttavia, questo approccio aggiunge costi e complessità aggiuntivi alla gestione di questo stato esterno e potrebbe non essere adatto per giochi frenetici o competitivi in cui i cambiamenti di stato sono così frequenti e avvengono su una scala così importante che l'introduzione anche di un potente archivio dati cache in memoria comporterebbe un ritardo di replica troppo significativo per essere

utile per ripristinare una sessione. Per giochi di questo tipo, l'approccio ottimale è accettare la perdita del server e rimandare il giocatore a una lobby di gioco per trovare un'altra sessione oppure puoi reindirizzarlo automaticamente a un'altra sessione di gioco.

Raccogli tutti i dati di registro utili su ciò che ha causato l'interruzione del server in modo da poter esaminare il problema in un secondo momento. Amazon GameLift fornisce indicazioni per il debug dei problemi della flotta e offre la possibilità di accedere in remoto alle istanze della flotta Amazon GameLift .

Passaggi dell'implementazione

- Monitora le metriche dei server di gioco per rilevare eventuali riduzioni delle prestazioni e rimuovi o sostituisci i server danneggiati se necessario per mantenere l'affidabilità.
- Usa Amazon ElastiCache o MemoryDB per gli aggiornamenti asincroni dello stato del gioco per abilitare il ripristino della sessione dopo i guasti del server, se possibile.
- Acquisisci dati di registro dettagliati sulle interruzioni dei server per indagini e debug, sfruttando strumenti come GameLift Amazon per il monitoraggio della flotta e l'accesso remoto.

GAMEREL03-BP02 Implementa un accoppiamento libero delle funzionalità di gioco per gestire i fallimenti con un impatto minimo sull'esperienza del giocatore

Il disaccoppiamento dei componenti si riferisce al concetto di progettazione dei componenti del server in modo che possano funzionare nel modo più indipendente possibile. Alcuni aspetti del gioco sono difficili da separare poiché i dati devono essere il più aggiornati possibile per offrire ai giocatori una buona esperienza di gioco. Tuttavia, molti componenti e attività di gioco possono essere disaccoppiati. Ad esempio, le classiifiche e i servizi di statistiche non sono fondamentali per l'esperienza di gioco e le letture e le scritture su questi servizi possono essere eseguite in modo asincrono dal gioco.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Implementa una riduzione graduale delle funzionalità del gioco, che possono essere disattivate automaticamente o da un amministratore se vengono rilevati problemi, oltre a configurare i servizi upstream che dipendono dalla funzionalità per poter gestire l'errore in modo corretto. Ad esempio, se i dati di un giocatore specifico non vengono caricati correttamente nel client di gioco, dovresti valutare

se questi dati sono fondamentali per l'esperienza di gioco. In caso contrario, configura il client di gioco in modo da gestire correttamente questo errore senza interrompere l'esperienza del giocatore, scegliendo di riprovare a recuperare questi dati in un secondo momento, quando il giocatore rivisita lo schermo.

Utilizza logiche come timeout, nuovi tentativi e backoff per gestire errori e fallimenti. I timeout impediscono ai sistemi di bloccarsi per periodi irragionevolmente lunghi. I nuovi tentativi possono garantire un'elevata disponibilità di errori transitori e casuali.

Definite i componenti non critici che possono essere associati liberamente ai componenti critici. L'accoppiamento flessibile consente ai sistemi di essere più resilienti poiché il guasto di un componente non si ripercuote a cascata sugli altri. Quando le funzionalità di gioco non richiedono connessioni stateful ai server di gioco o al backend, dovresti implementare protocolli stateless per scalare dinamicamente e ripristinare guasti transitori. Sviluppa i componenti non critici laddove possano essere associati liberamente a protocolli stateless utilizzando un'API. HTTP/JSON Implementa le chiamate di rete dal client di gioco in modo che siano asincrone e non bloccanti per ridurre al minimo l'impatto sui giocatori delle funzionalità di gioco a prestazioni lente o di altri servizi dipendenti.

Per migliorare ulteriormente la resilienza tramite l'accoppiamento libero, utilizza un servizio di messaggistica come un sistema di accodamento, streaming o un sistema basato su argomenti tra componenti che possono essere gestiti in modo asincrono. Questo modello è adatto per un'interazione che non richiede una risposta immediata o in cui è sufficiente confermare che una richiesta è stata registrata. Questa soluzione prevede un componente che genera eventi e un altro che li consuma. I due componenti non si integreranno tramite un' point-to-pointinterazione diretta ma attraverso un elemento intermedio, ad esempio uno storage durevole o uno strato di coda. Questo aiuta anche a migliorare l'affidabilità del sistema preservando i messaggi quando l'elaborazione fallisce.

Ricerca e seleziona un meccanismo di messaggistica appropriato, poiché vari servizi di messaggistica hanno caratteristiche diverse, come i meccanismi di ordinazione e consegna. Progetta le operazioni in modo che siano idempotenti in modo che il sistema di messaggi scelto recapiti i messaggi almeno una volta. Ad esempio, prendiamo in considerazione un tipico caso d'uso di un gioco in cui il gioco deve tenere traccia del tempo di gioco dei giocatori, delle statistiche o di altri dati pertinenti, il che può portare a un elevato throughput di scrittura nei momenti di picco della concorrenza tra i giocatori.

Per implementare un'architettura affidabile, valuta se il caso d'uso richieda la read-after-write coerenza percepita dal giocatore. In genere, scenari come questi sono adatti all'elaborazione

asincrona e possono essere realizzati implementando un modello di coda di scrittura in cui le richieste vengono inserite in una coda di messaggi scalabile e durevole come Amazon SQS e possono essere inserite nel database di backend in batch utilizzando un servizio consumer, come una funzione Lambda. Questo approccio è più affidabile della comunicazione sincrona tra più componenti distribuiti, tra cui il client di gioco del giocatore, i server web e applicativi di backend e il sistema di database interno. Inoltre, riduce i costi perché non è necessario scalare il database di backend per soddisfare i picchi di velocità di scrittura, poiché l'elaborazione da parte dei consumatori dalla coda di scrittura può essere utilizzata per rallentare questa velocità di acquisizione in base alle esigenze.

Passaggi dell'implementazione

- Separa i componenti non critici come le classifici e i servizi di statistiche dalle funzionalità di gioco fondamentali per consentire operazioni asincrone e migliorare la resilienza.
- Implementa una riduzione graduale delle funzionalità non critiche con una logica per timeout, nuovi tentativi e backoff, e verifica che il client di gioco gestisca gli errori senza interrompere l'esperienza del giocatore.
- Utilizza sistemi di messaggistica come Amazon SQS per la comunicazione asincrona tra i componenti, che consentono l'elaborazione scalabile, duratura e affidabile di casi d'uso con throughput elevato.

Resources

- Crea carichi di lavoro altamente scalabili e affidabili utilizzando l'architettura a microservizi
- Integrazione di microservizi utilizzando servizi serverless AWS
- Comprendere la messaggistica asincrona per i microservizi
- Introduzione ai modelli di sviluppo di giochi scalabili su AWS
- Implementazione di Graceful Degradation

GAMEREL03-BP03 Monitora gli eventi dell'infrastruttura nel tempo per misurare l'impatto sul comportamento dei giocatori

Monitora il processo del server di gioco, le metriche delle istanze del server di gioco e le metriche dell'esperienza di gioco per determinare la causa principale dei problemi. Oltre al monitoraggio della CPU e della memoria, puoi anche configurare il monitoraggio delle metriche di rete relative alle limitazioni di rete delle EC2 istanze per avvisarti di problemi come l'eccesso di larghezza di banda

o altri problemi a livello di rete che packets-per-second potrebbero indicare che le risorse del server non sono sufficienti.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Usa CloudWatch Synthetics per verificare le funzionalità dell'applicazione fondamentali per l'esperienza del giocatore, come l'impossibilità di accedere o altri problemi che influiscono sul servizio. Per i server di gioco ospitati su Amazon GameLift, valuta la possibilità di monitorare parametri come:

- GameServerInterruptions e InstanceInterruptions, questo può aiutarti a capire in che modo le limitazioni della disponibilità delle istanze Spot influiscono sui server di gioco distribuiti con Spot.
- ServerProcessAbnormalTerminations, che può essere utilizzato per rilevare interruzioni anomale nei processi del server di gioco.

Si consiglia di conservare i dati storici delle metriche relative all'affidabilità del server di gioco. Utilizza questi dati storici per scopi di reportistica e uniscili ad altri set di dati per scoprire potenziali tendenze e valutare l'impatto sul comportamento dei giocatori nel tempo che potrebbero essere dovuti a problemi dei server di gioco.

Amazon CloudWatch non conserva i parametri a tempo indeterminato e la risoluzione di archiviazione dei parametri aumenta nel tempo, quindi valuta la possibilità di esportare questi parametri in sistemi di storage a lungo termine convenienti come Amazon S3. Puoi configurare CloudWatchMetric Streams per distribuire automaticamente i parametri dalle CloudWatchregioni al tuo bucket S3, dove possono essere archiviati a lungo termine in un livello di storage come S3 Intelligent-Tiering e infine archiviati utilizzando Amazon Glacier. Inserendo le tue metriche in Amazon S3, sono subito disponibili per essere unite ad altri set di dati nel tuo data lake per l'interrogazione interattiva con Amazon Athena.

Passaggi dell'implementazione

- Monitora i parametri del server di gioco, delle istanze e della rete, tra cui larghezza di banda e packet-per-second limiti, utilizzando Amazon e CloudWatch Synthetics per i controlli delle CloudWatch funzionalità dei percorsi critici.
- Tieni traccia GameLift di metriche specifiche come GameServerInterruptionse ServerProcessAbnormalTerminationsvaluta l'impatto della disponibilità delle istanze Spot e rileva terminazioni anomale dei server.

- Esporta i CloudWatch parametri in Amazon S3 per lo storage a lungo termine, usa livelli convenienti come S3 Intelligent-Tiering o Glacier e analizza le tendenze con strumenti come Amazon Athena.

Resources

- Le metriche delle prestazioni di rete EC2 a livello di istanza di Amazon rivelano nuove informazioni
- CloudWatchMetric Streams: invia i AWS parametri ai partner e alle tue app in tempo reale

Resources

Consulta le seguenti risorse per saperne di più sulle nostre best practice relative all'affidabilità.

Documenti correlati:

- Praticare l'integrazione continua e la distribuzione continua su AWS
- Scalabilità automatica delle code di lavoro asincrone
- Progetta la tua architettura WorkloadService
- Timeout, nuovi tentativi e backoff con jitter
- Well-Architected Framework - Pilastro dell'affidabilità
- Progettazione per una scalabilità affidabile
- La libreria di Amazon Builder
- Messaggistica in tempo reale su vasta scala per giochi multigiocatore
- Introduzione ai modelli di sviluppo di giochi scalabili su AWS
- Esecuzione di microservizi containerizzati su AWS
- Hosting di applicazioni Web nel cloud
- Creazione di un'infrastruttura di rete multi-VPC scalabile e sicura

Video correlati:

- re:Invent 2020: Ubisoft: creazione di un gioco multiplayer multipiattaforma su AWS
- re:Invent 2018: Supercell - Giochi mobili in scala
- re:Invent 2019: Come CAPCOM crea giochi divertenti con contenitori, dati e apprendimento automatico

- re:Invent 2018: globalizzazione degli account dei giocatori in Riot Games mantenendo al contempo la disponibilità
- re:Invent 2020: GameLoft - Approfondimento sulla migrazione ai data lake con zero downtime

Formazione correlata:

- Utilizzo di Amazon GameLiftFleet IQ per server di gioco
- Hosting di server di gioco con Amazon EC2

Efficienza delle prestazioni

Il pilastro dell'efficienza delle prestazioni si concentra sull'uso efficiente delle risorse di elaborazione per soddisfare i requisiti e sul mantenimento di tale efficienza man mano che la domanda cambia e le tecnologie si evolvono.

Adotta un approccio basato sui dati per selezionare un'architettura ad alte prestazioni. Raccogli dati completi sull'architettura, dalla progettazione di alto livello alla selezione e configurazione dei tipi di risorse. Valuta le tue scelte architettoniche su base ciclica per sfruttare il set di servizi e soluzioni in continua evoluzione. Le metriche aiutano a comprendere la deviazione dalle prestazioni previste in modo da poter agire. Un approccio basato sui dati aiuta a trovare compromessi con l'architettura per migliorare le prestazioni, ridurre i costi o migliorare l'esperienza degli sviluppatori.

Aree di interesse

- Principi di progettazione
- Scelta dell'architettura
- Selezione della regione
- Sviluppo iterativo
- Calcolo e hardware
- Calcola la selezione
- Gestione dei dati
- Reti e distribuzione di contenuti
- Processo e cultura
- Resources

Principi di progettazione

Oltre ai principi di progettazione contenuti nel white paper AWS Well-Architected Framework, i seguenti principi di progettazione possono garantire l'efficienza delle prestazioni dei tuoi giochi:

- Misura le prestazioni di gioco partendo da end-to-end: È importante misurare le prestazioni così come vengono percepite dal punto di vista dei giocatori. Ciò significa che dovresti misurare le prestazioni del client di gioco, dell'infrastruttura di gioco e della connettività Internet che collega i giocatori all'infrastruttura. Ciò contribuirà a capire dove è possibile apportare miglioramenti delle prestazioni all'interno della propria architettura.

- Ottimizza la tua architettura per migliorare le metriche che riflettono l'esperienza effettiva del giocatore: man mano che adatti ed evolvi la tua architettura nel tempo, pensa a come questi miglioramenti e modifiche influiranno sull'esperienza del giocatore. I carichi di lavoro legati ai giochi dovrebbero essere in grado di resistere e ridurre al minimo l'impatto dei guasti per bloccare le interruzioni diffuse del gameplay. Le funzionalità e i sistemi di gioco che non dipendono in modo critico l'uno dall'altro devono essere disaccoppiati per ridurre il raggio d'azione dei guasti e isolare i problemi che influiscono sui giocatori.
- Utilizza tecnologie che semplificano le operazioni di gioco e aumentano la velocità di sviluppo: dai la priorità all'adozione di tecnologie in grado di migliorare l'efficienza degli sviluppatori. Il sovraccarico operativo durante le fasi di sviluppo di pre-produzione può distrarre dal miglioramento del gameplay. Sfruttando i servizi gestiti dei AWS nostri partner, è possibile ridurre i costi ingegneristici, il che consente agli sviluppatori di giochi di concentrarsi sul ciclo di gioco principale e sull'esperienza del giocatore. AWS I requisiti di architettura e prestazioni possono cambiare ed evolversi durante il ciclo di vita dello sviluppo del gioco e in ogni fase è necessario prendere in considerazione i compromessi tecnologici.
- Progetta l'infrastruttura per soddisfare il picco di concorrenza tra i giocatori e scalala dinamicamente in base alle esigenze: l'infrastruttura deve essere progettata in modo da scalare per soddisfare la domanda dei giocatori. Le metriche, come la concomitanza delle sessioni dei giocatori e il numero di accessi, possono essere utilizzate per scalare preventivamente prima che i sistemi si sovraccarichino. Le metriche di utilizzo reattivo del sistema, come il consumo di CPU e memoria, possono essere utilizzate per scalare dopo il sovraccarico dei sistemi. Scalando dinamicamente l'infrastruttura, puoi ridurre i costi operativi del gioco.

Scelta dell'architettura

GAMEPERF01: Come selezionate l'opzione di hosting appropriata per i vostri server di gioco?

La selezione dell'opzione di hosting appropriata per i tuoi server di gioco è fondamentale per le prestazioni del server di gioco. Decidere di utilizzare EC2 istanze, una soluzione container o un servizio completamente gestito è una delle prime decisioni da prendere quando si progetta per la produzione. Ogni opzione di hosting avrà funzionalità e considerazioni diverse per l'ottimizzazione delle prestazioni, la scalabilità, le operazioni e l'integrazione.

Best practice

- GAMEPERF01-BP01 Valuta i requisiti di risorse del server di gioco e le esigenze di scalabilità
- GAMEPERF01-BP02 Prendi in considerazione il sovraccarico operativo per scalare i server di gioco
- GAMEPERF01-BP03 Valuta l'integrazione con altri AWS servizi, ambienti di sviluppo, architetture CPU target e funzionalità

GAMEPERF01-BP01 Valuta i requisiti di risorse del server di gioco e le esigenze di scalabilità

Valuta i requisiti del server rispetto alle tue esigenze di scalabilità per verificare che stai selezionando un'opzione di hosting che soddisfi i tuoi requisiti e offra prestazioni ottimali.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Quando scegli l'opzione di hosting appropriata per i tuoi server di gioco, considera i seguenti fattori:

Requisiti delle risorse del server di gioco

Valuta i requisiti di CPU, memoria, rete e archiviazione dei processi del server di gioco per determinare il consumo del gioco. Non trascurate il networking: ogni frame richiede cicli di CPU per ricevere le azioni del giocatore, aggiornare lo stato del gioco e inviarlo al giocatore. L'offloading dell'elaborazione dei pacchetti può liberare la CPU per le funzioni principali del gioco. Il networking è la base per un gioco fluido e reattivo, quindi testarlo nelle prime fasi del processo definisce un profilo prestazionale di base per un gioco.

Uno sparatutto in prima persona potrebbe avere elevate azioni al secondo necessarie alla CPU per spostarsi rapidamente sulla rete, il che potrebbe favorire le istanze della famiglia C ottimizzate per il calcolo, mentre un gioco di strategia a turni che può impiegare più cicli di CPU per turno potrebbe richiedere una maggiore memoria dalle istanze della famiglia R per archiviare e aggiornare localmente lo stato del gioco sul server prima di rimandarlo ai giocatori. Utilizza un approccio basato sui dati come il metodo Utilization Saturation and Errors (USE) per fare scelte architettoniche ben informate.

Scalabilità ed elasticità

Valuta la rapidità e la facilità con cui ogni opzione di hosting può scalare per soddisfare la domanda dei giocatori senza compromettere le prestazioni. Considera il livello di automazione e flessibilità

richiesto per il carico di lavoro del tuo gioco per mantenere un'esperienza di gioco fluida durante le ore di punta. Un server di gioco potrebbe scalare rapidamente aumentando l'utilizzo aggiungendo processi di server di gioco aggiuntivi sulla stessa istanza, mentre un backend di gioco può scalare più lentamente in base all'aumento del numero di utenti attivi e dei giochi giocati. La tua flotta dovrebbe adattarsi alla domanda per ridurre al minimo i costi e allo stesso tempo ridurre al minimo i tempi di attesa dei giocatori per entrare in gioco. Consulta Amazon EC2 Spot Instance Advisor per ottenere informazioni sulla capacità disponibile a costi contenuti per le flotte di server di gioco.

Passaggi dell'implementazione

- Valuta i requisiti di risorse del server di gioco in termini di CPU, memoria, rete e storage per selezionare i tipi di istanze più adatti, tenendo conto delle esigenze prestazionali specifiche del gioco, come l'elevata velocità di rete per i giochi FPS o l'ottimizzazione della memoria per i giochi di strategia a turni.
- Confronta diverse opzioni di hosting come contenitori, istanze, bare-metal e servizi gestiti analizzando i dati sulle prestazioni utilizzando framework come il metodo USE. Usa queste informazioni per prendere decisioni migliori sull'architettura del tuo sistema.
- Progetta flotte che garantiscano scalabilità ed elasticità, sfruttando strumenti come EC2 Spot Instance Advisor per ottimizzare i costi e facilitare al contempo una scalabilità rapida per soddisfare la domanda dei giocatori nelle ore di punta.

GAMEPERF01-BP02 Prendi in considerazione il sovraccarico operativo per scalare i server di gioco

Considerate il sovraccarico gestionale e operativo associato a ciascuna opzione di hosting.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Sovraccarico operativo

Le soluzioni ospitate autonomamente sui EC2 nostri container possono fornire un maggiore controllo, ma richiedono anche una maggiore gestione. Gli orchestratori di container come ECS o EKS possono ridurre i tempi di avvio dei server containerizzati, aumentando al contempo la complessità della rete e il sovraccarico di orchestrazione della manutenzione.

Ad esempio, i gruppi di nodi gestiti da EKS possono automatizzare il provisioning e la gestione del ciclo di vita dei server di gioco, ma non rispettano i budget per le interruzioni dei pod quando si chiude

un nodo. Se il gioco richiede un periodo di terminazione superiore a 15 minuti per completare i giochi in sicurezza, potrebbe essere necessario creare hook del ciclo di vita o prendere in considerazione nodi autogestiti con controller personalizzati per bloccare l'interruzione del gioco.

I servizi gestiti come Amazon Game Lift possono gestire la maggior parte del sovraccarico operativo, ma ridurre la visibilità e il controllo sui requisiti speciali per la configurazione di rete e sicurezza di basso livello. La scelta di una soluzione per server di gioco è un compromesso tra il livello di personalizzazione, il controllo e la responsabilità che avrai nell'ottimizzare le prestazioni del server di gioco e il comportamento di scalabilità.

Passaggi dell’implementazione

- Valuta il sovraccarico operativo per le opzioni di hosting, bilanciando gli sforzi di controllo e gestione tra soluzioni self-hosted come EC2 ECS o EKS e servizi gestiti come Amazon Game Lift.
- Usa i gruppi di nodi gestiti da EKS per l'automazione, ma implementa i lifecycle hook o i controller personalizzati se i tuoi server di gioco richiedono periodi di terminazione più lunghi rispetto a quelli predefiniti.
- Valuta i compromessi tra personalizzazione, visibilità e responsabilità operativa nella scelta di una soluzione per server di gioco.

GAMEPERF01-BP03 Valuta l'integrazione con altri AWS servizi, ambienti di sviluppo, architetture CPU target e funzionalità

Valuta la capacità di integrazione di ciascuna opzione di hosting con altri AWS servizi su cui si basa il gioco, come database, analisi o servizi di distribuzione di contenuti.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Integrazione con altri servizi AWS

La perfetta integrazione tra i servizi offre vantaggi operativi come un migliore monitoraggio delle prestazioni e una distribuzione efficiente e sicura dei dati tra componenti di gioco, server di gioco, servizi di backend di gioco e soluzioni di osservabilità.

Ad esempio, coordinare i turni di traffico per le partite dal vivo può essere complesso. Amazon Route 53 ti aiuterà a mantenere aggiornati i tuoi record DNS, semplificando gli spostamenti coordinati del

traffico. AWS Le ghiere di traffico di Global Accelerator ti consentono di inviare una percentuale di traffico verso un'altra regione e di mantenere il gioco attivo durante la manutenzione.

Ambiente e strumenti di sviluppo

Prendi in considerazione gli strumenti, i framework e gli ambienti di sviluppo supportati da ciascuna opzione di architettura. Verifica che l'opzione scelta sia in linea con la soluzione di sviluppo del gioco e con i linguaggi di programmazione, poiché ciò può influire sulla capacità del team di ottimizzare e mantenere le prestazioni del server di gioco. La distribuzione di un gioco su dispositivi mobili, console e PC aumenterà la complessità degli strumenti e dei test. Il supporto tra sistemi è particolarmente importante per gli studi che utilizzano più giochi, in cui i servizi centralizzati possono standardizzare le migliori pratiche di sviluppo per tutti i titoli.

Architettura e funzionalità della CPU Target

Considerate il profilo prestazionale del motore di gioco e dei processi del server di gioco e il livello di supporto ARM disponibile. Valuta se puoi trarre vantaggio dal miglioramento del rapporto prezzo/prestazioni dei processori Graviton basati su ARM o x86. AMD64 Devi usare funzionalità Intel come la crittografia AES-NI, AVX o Turbo Boost? Esamina i tipi di host dedicati per identificare le famiglie di istanze a socket singolo o multiplo. Quando utilizzi una famiglia di istanze multi-socket, prendi in considerazione il pinning NUMA e la condivisione della cache L3 nei processi del server di gioco. Usa la configurazione C-state e P-state per ottenere le migliori prestazioni per il tuo gioco regolando la frequenza di clock e riducendo i livelli di sonno.

Passaggi dell’implementazione

- Seleziona le opzioni di hosting con perfetta integrazione con AWS servizi come Gestione dei segreti AWS ACM e altri per semplificare il monitoraggio delle prestazioni, proteggere la consegna dei dati e ridurre le attività operative manuali.
- Verifica la compatibilità tra la tua opzione di hosting e l'ambiente di sviluppo, i framework e i linguaggi di programmazione per ottimizzare e mantenere le prestazioni del server in modo efficace.
- Valuta i requisiti dell'architettura della CPU, sfruttando Graviton per il rapporto prezzo/prestazioni o x86 per funzionalità specifiche come AES-NI, AVX e Turbo Boost, e ottimizza le prestazioni del server con il pinning NUMA e l'ottimizzazione C-state/p-state.

Selezione della regione

GAMEPERF02: Come determini le regioni geografiche in cui ospitare la tua infrastruttura di gioco?

La scelta della posizione ideale per la tua infrastruttura di gioco può migliorare le prestazioni di rete per i giocatori e il backend. Considerare da dove si connette la tua base di giocatori e come sono costruiti le tue community o i tuoi server è importante per la crescita e la sostenibilità a lungo termine in una regione geografica. L'implementazione di un'infrastruttura di server di gioco e di servizi di backend disaccoppiati può favorire l'efficienza operativa complessiva e migliorare la resilienza utilizzando più regioni, zone locali e avamposti per ospitare il gioco.

Best practice

- GAMEPERF02-BP01 Seleziona una regione d'origine vicina ai tuoi giocatori
- GAMEPERF02-BP02 Progetta un approccio che supporti la collocazione di infrastrutture di gioco sensibili alla latenza vicino ai giocatori per migliorare le prestazioni

GAMEPERF02-BP01 Seleziona una regione d'origine vicina ai tuoi giocatori

Per il lancio iniziale del gioco, dovresti decidere dove installare l'infrastruttura sulla base di discussioni con gli stakeholder aziendali, ad esempio i team di pubblicazione che stabiliscono dove il gioco dovrebbe essere reso disponibile ai giocatori e dove concentrare le proprie attività di marketing e pubblicità prima del lancio.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

I vostri stakeholder aziendali dovrebbero inoltre disporre di meccanismi atti a stimolare la domanda, in modo da comprendere meglio l'accoglienza e la redditività dei giocatori. Ad esempio, queste squadre disporranno di meccanismi quali preordini di giochi, eventi e campagne di marketing, elenchi di e-mail pubblici per consentire ai giocatori di manifestare interesse prima del lancio e altri approcci per stabilire segnali pertinenti per determinare dove il gioco avrà probabilmente il maggior numero di giocatori al momento del lancio. Il gioco può anche utilizzare una strategia di lancio regionale che include fasi di play test e soft launch per determinare la domanda dei giocatori a livello regionale.

Seleziona una regione d'origine vicina alla tua base di giocatori e ai tuoi sviluppatori e che offra AWS i servizi e le funzionalità di cui hai bisogno per ospitare il gioco. La home page RSegion sarà il luogo in cui verranno eseguiti i servizi di backend del gioco e potrebbe anche eseguire i server di gioco. Valuta una regione di origine in base ai servizi supportati, alla connettività alle edge location, alla vicinanza alle regioni di failover e al numero di zone di disponibilità. Se utilizzi una zona locale, considera che la regione principale a volte si trova in un'area geografica diversa. Ad esempio: Santiago, Cile La zona locale us-east-1-scl-1a ha come regione madre la Virginia settentrionale us-east-1 anche se geograficamente è più vicina alla sa-east-1 di San Paolo.

Passaggi dell’implementazione

- Identifica le regioni di distribuzione in base ai segnali di domanda dei giocatori derivanti da attività pre-lancio come preordini, campagne di marketing e registrazione degli interessi.
- Scegli una regione di residenza vicina alla base di giocatori e agli sviluppatori principali, assicurandoti che supporti i AWS servizi richiesti, le edge location e le regioni di failover.
- Valuta attentamente le Local Zones, considerando che la Regione madre può differire geograficamente dall'ubicazione della Local Zone.

GAMEPERF02-BP02 Progetta un approccio che supporti la collocazione di infrastrutture di gioco sensibili alla latenza vicino ai giocatori per migliorare le prestazioni

Il posizionamento separato per infrastrutture sensibili alla latenza come i server di gioco riduce al minimo l'impatto dei lunghi percorsi di rete. Le implementazioni ripetibili possono semplificare la gestione di più postazioni più performanti per i giocatori. Il ping è una metrica comune che compare nell'interfaccia utente di gioco e un ping basso può essere una caratteristica distintiva.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Quando avvii un gioco per la prima volta, potresti non disporre ancora di informazioni sufficienti sulla tua base di giocatori per sapere in modo adeguato dove implementare l'infrastruttura più vicina ai giocatori più interessati a giocare al tuo gioco. Si tratta di una sfida comune e dovreste prepararvi a questo scenario progettando un'architettura che consenta di adattare rapidamente la vostra strategia di posizionamento dell'hosting per distribuire i server dove sono necessari più vicini ai giocatori. È normale che gli sviluppatori di giochi valutino regolarmente l'implementazione dell'infrastruttura

di gioco come analisi periodica post-lancio per investire in modo incrementale in miglioramenti nel tempo con un approccio iterativo.

Una best practice consiste nell'utilizzare infrastructure-as-code modelli, come AWS CloudFormation o Terraform di Hashicorp, per la configurazione dell'infrastruttura VPCs, ad esempio le configurazioni di sottorete e le dipendenze necessarie per avviare servizi di gioco critici in modo da poter fare riferimento a questi modelli, personalizzarli rapidamente se necessario e distribuirli in luoghi in cui è necessaria un'infrastruttura aggiuntiva per supportare i giocatori.

È inoltre necessario assicurarsi di comprendere in che modo la strategia di implementazione attuale potrebbe essere evoluta per consentire future espansioni. I modelli IaC sono ripetibili ma non sostituiscono la pianificazione della rete. IPAM gestisce il tuo. VPCs Dimensionamento della sottorete, selezione della zona di disponibilità, inventario IP e allineamento della zona di disponibilità tra account. La rete è importante da prendere in considerazione e può causare problemi ai giocatori quando viene modificata. I server di gioco distribuiti in più aree geografiche si collegheranno al backend di gioco, che è più comune se ospitati in una o più aree geografiche, che possono richiedere una configurazione aggiuntiva per supportare la connettività privata. Queste considerazioni devono essere valutate continuamente nel tempo in modo da poter apportare modifiche alla strategia di hosting dei giochi man mano che i requisiti del gioco evolvono o cambiano le esigenze dei giocatori.

Nel determinare il numero di sedi di hosting dei giochi da utilizzare per il gioco, considera i seguenti fattori:

- Miglioramento della qualità dell'esperienza dei giocatori: qual è l'entità del miglioramento dell'esperienza utente che puoi apportare aggiungendo altre sedi di hosting dei giochi? Qual è l'aumento incrementale delle prestazioni che puoi ottenere in questo modo? Come misurerai questo miglioramento delle prestazioni?
- A quali gruppi di giocatori dare priorità: per quanti giocatori puoi migliorare l'esperienza aggiungendo altre sedi di hosting dei giochi? A quali popolazioni di giocatori, o località geografiche, darai priorità?
- Impatti a valle del cambiamento: se modifichi la tua strategia di hosting dei giochi, in che modo ciò influirà sui tempi di attesa dei giocatori nel matchmaking? Le dimensioni delle partite, il bilanciamento delle abilità o il numero di giocatori nel pool di giocatori possono far fronte a un cambiamento di strategia in materia di location che ospita la partita? Supportare più sedi può potenzialmente frammentare il pool di giocatori e aumentare costi e complessità.

Ciascuna di queste considerazioni dovrebbe essere valutata nel determinare dove aggiungere o rimuovere le sedi di hosting dei giochi. Ad esempio, puoi scegliere di dare priorità al miglioramento dell'esperienza per i giocatori in aree geografiche con l'esperienza di gioco meno performante o per i giocatori che esprimono il feedback pubblico più esplicito. Puoi anche scegliere di inserire tra le tue priorità la monetizzazione dei giocatori, ad esempio concentrando l'attenzione sul miglioramento dell'esperienza dei giocatori in aree geografiche che generano una fonte significativa di entrate per il tuo gioco o che hanno il potenziale di generare entrate incrementali se introduci miglioramenti delle prestazioni.

Oltre all'infrastruttura di hosting in Regioni AWS, puoi utilizzare Local Zones, che sono un'estensione di un Regione AWS, per ospitare i tuoi server di gioco e altre applicazioni sensibili alla latenza come i server di chat vocale più vicini ai tuoi giocatori. Puoi anche scegliere di gestire l'infrastruttura di sviluppo dei giochi in Local Zones per migliorare l'esperienza dei tuoi team di sviluppo di giochi. Ad esempio, puoi utilizzare Local Zones per risolvere casi d'uso come l'hosting di repliche dei tuoi server di controllo del codice sorgente autogestiti più vicino agli sviluppatori di giochi e per offrire workstation virtuali per lo sviluppo di giochi e storage di contenuti agli utenti che utilizzano EC2 istanze Amazon, volumi EBS e FSx file system Amazon distribuiti in una o più Local Zones vicino ai tuoi studi di sviluppo senza richiedere l'hosting dell'infrastruttura in locale.

Gli Outposts sono un'ottima scelta quando le Regioni o le Local Zones non sono disponibili nella stessa area geografica. La connettività dal data center a AWS dovrebbe essere presa in considerazione per garantire l'affidabilità del server di gioco rispetto al sistema di backend. AWS Outposts e Outpost Servers sono progettati appositamente per funzionare AWS nel tuo datacenter utilizzando gli stessi servizi e APIs per contribuire a creare un modello di distribuzione coerente ovunque esegui il gioco. È possibile combinare più rack in un Outpost logico e condividere l'infrastruttura. Account AWS Il ciclo di vita dell'hardware è gestito da AWS e il lead time può essere di soli 3 mesi.

Se stai creando giochi utilizzando contenitori e desideri la flessibilità necessaria per adottare un'architettura di distribuzione ibrida utilizzando software open source che può essere distribuito sulla tua infrastruttura locale, puoi utilizzare ECS Anywhere o EKS Anywhere come alternativa alle nostre Local Zones. AWS Outposts Se offri hosting con Amazon GameLift, Amazon GameLift Anywhere può essere usato per far funzionare il tuo server su hardware locale, velocizzando il processo di sviluppo, consentendoti di utilizzare Local Zones o registrare il tuo metallo come parte del tuo parco macchine.

Passaggi dell'implementazione

- Usa infrastructure-as-code strumenti come AWS CloudFormation Terraform per implementazioni ripetibili, che consentono la personalizzazione e la scalabilità rapide delle sedi di hosting dei giochi in base alle esigenze dei giocatori.
- Valuta i miglioramenti dell'esperienza dei giocatori, le priorità della popolazione di giocatori e gli impatti a valle, come i tempi di matchmaking, quando aggiungi o rimuovi sedi che ospitano i giochi.
- Utilizza AWS Local Zones, Outposts o opzioni ibride come ECS Anywhere, EKS Anywhere o GameLift Anywhere per ottimizzare l'infrastruttura sensibile alla latenza e supportare diverse esigenze di implementazione.

Sviluppo iterativo

GAMEPERF03: Come puoi usare Amazon GameLift per l'efficienza delle prestazioni di sviluppo iterativo?

Amazon GameLift fornisce un end-to-end flusso di lavoro per lo sviluppo e il test delle prestazioni del gioco in un ambiente di test locale.

Best practice

- GAMEPERF03-BP01 Usa Amazon GameLift Anywhere e un toolkit di test GameLift
- GAMEPERF03-BP02 Verifica le prestazioni e la scalabilità dei server di gioco
- GAMEPERF03-BP03 Ottimizza l'utilizzo delle risorse dei contenitori GameLift

GAMEPERF03-BP01 Usa Amazon GameLift Anywhere e un toolkit di test GameLift

Per migliorare l'efficienza delle prestazioni attraverso un processo di sviluppo iterativo, utilizza Amazon GameLift Anywhere insieme ad Amazon GameLift Testing Toolkit per creare un ambiente di test completo.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Questo approccio consente un'iterazione rapida, una raccolta efficiente dei dati e un'analisi dettagliata delle prestazioni. I passaggi chiave includono:

Creare un ambiente di test

Usa Amazon GameLift Anywhere per configurare un ambiente di test locale o basato sul cloud. Questa configurazione elimina la necessità di caricare ogni iterazione di build del server di gioco su una flotta gestita, riducendo i tempi di attivazione.

Integra Amazon GameLift Testing Toolkit

Incorpora Amazon GameLift Testing Toolkit nel tuo flusso di lavoro di sviluppo. Il toolkit fornisce script, strumenti e librerie per visualizzare l'infrastruttura GameLift Amazon, avviare player virtuali e iterare FlexMatch sui set di regole con il simulatore. FlexMatch Semplifica l'integrazione e la gestione delle GameLift risorse Amazon, consentendoti di automatizzare le attività comuni e raccogliere i dati necessari per l'analisi delle prestazioni.

Cicli rapidi di costruzione e test

Aggiorna rapidamente la flotta di test con nuove build, avviala e inizia i test. Questo facilita un build-test-repeat ciclo rapido, permettendo agli sviluppatori di convalidare vari aspetti dell'esperienza di gioco, incluse le interazioni multigiocatore.

Test completi

Testa l'integrazione del tuo server di gioco con l'SDK del GameLift server Amazon, le interazioni dei servizi di backend, le configurazioni di matchmaking e altre funzionalità di hosting. GameLift Utilizza il GameLift Testing Toolkit per automatizzare i test e raccogliere metriche dettagliate sulle prestazioni, assicurandoti che i componenti del gioco funzionino perfettamente insieme.

Analizza i dati sulle prestazioni

Usa i dati raccolti dal GameLift Testing Toolkit per analizzare i problemi di prestazioni e ottimizzare il tuo server di gioco. Il toolkit aiuta a tenere traccia delle metriche chiave, a identificare i problemi e a prendere decisioni basate sui dati per migliorare l'efficienza delle prestazioni.

Incorporando Amazon GameLift Anywhere e GameLift Testing Toolkit nel tuo processo di sviluppo iterativo, puoi migliorare in modo significativo l'efficienza delle prestazioni attraverso test rapidi, controlli di integrazione completi e analisi dettagliate delle prestazioni.

Passaggi dell’implementazione

- Usa Amazon GameLift Anywhere per creare un ambiente di test, riducendo i tempi di attivazione per le build dei server di gioco e consentendo un'iterazione rapida.
- Integra Amazon GameLift Testing Toolkit per automatizzare le attività di test, simulare giocatori e convalidare FlexMatch le configurazioni durante lo sviluppo.
- Raccogli e analizza i dati sulle prestazioni con GameLift Testing Toolkit per identificare i punti deboli, ottimizzare i server di gioco e migliorare l'efficienza delle prestazioni.

GAMEPERF03-BP02 Verifica le prestazioni e la scalabilità dei server di gioco

Per testare le prestazioni e la scalabilità dei tuoi server di gioco, implementa un solido framework di test utilizzando le GameLift funzionalità di Amazon e il GameLift Testing Toolkit.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Le pratiche chiave includono:

Test iterativi

Usa una flotta Amazon GameLift Anywhere per creare un ambiente ospitato basato sul cloud in cui creare e testare i componenti di gioco in modo iterativo. Questo ambiente dovrebbe rispecchiare le condizioni di hosting del mondo reale, consentendo test realistici di prestazioni e scalabilità.

Test di integrazione dei server di gioco

Verifica l'integrazione del tuo server di gioco con l'SDK del GameLift server Amazon, incluso l'avvio di nuove sessioni di gioco e il monitoraggio degli eventi delle sessioni di gioco utilizzando AWS CLI il nostro GameLift Testing Toolkit. Ciò verifica che il server di gioco funzioni correttamente all'interno dell'ambiente. GameLift

Utilizza il GameLift Testing Toolkit per automatizzare i test e raccogliere metriche dettagliate sulle prestazioni. Il toolkit consente di visualizzare l' GameLift infrastruttura, avviare player virtuali per i test di carico e iterare sui FlexMatch set di regole con il simulatore. FlexMatch È particolarmente utile per scalare le attività di ECS Fargate, che simula le sessioni dei giocatori creando numerose sessioni di gioco simultanee per testare lo stress dell'infrastruttura del server.

Test di scalabilità

Sperimenta con la progettazione di code per sessioni di gioco, flotte con più sedi, flotte Spot e On-Demand e più tipi di istanze. Prova le opzioni di posizionamento delle sessioni di gioco, le politiche di latenza e le impostazioni di prioritizzazione della flotta. Configura la scalabilità della capacità per soddisfare la domanda dei giocatori e verifica che il sistema sia in grado di gestire il carico previsto in condizioni diverse.

Passaggi dell’implementazione

- Usa Amazon GameLift Anywhere per configurare un ambiente di test realistico per test iterativi di prestazioni e scalabilità.
- Testa l'integrazione del server di gioco con l'SDK del GameLift server, facilitando la corretta gestione delle sessioni e il monitoraggio degli eventi all'interno dell'ambiente. GameLift
- Esegui test di scalabilità con Testing Toolkit, simulando il carico GameLift dei giocatori, testando le code delle sessioni e convalidando il ridimensionamento della flotta, le politiche di latenza e le impostazioni di prioritizzazione.

GAMEPERF03-BP03 Ottimizza l'utilizzo delle risorse dei contenitori GameLift

Per ottimizzare l'utilizzo delle risorse dei GameLift container, progetta la tua flotta di container in modo efficace e stabilisci limiti precisi delle risorse.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Le linee guida principali includono:

- Progettazione di gruppi di contenitori: organizza il software in gruppi di contenitori. Il contenitore principale dovrebbe raggruppare l'applicazione del server di gioco e Amazon GameLift Agent. Utilizza i contenitori sidecar per aggiungere software per gestire le dipendenze e impostare limiti specifici per l'utilizzo di memoria e CPU.
- Imposta i limiti delle risorse: per ogni gruppo di contenitori, determina le risorse di memoria e CPU richieste. Imposta limiti opzionali per i singoli contenitori per verificare che abbiano risorse riservate, ma puoi anche superarli se sono disponibili risorse aggiuntive. Questo aiuta a prevenire la contesa delle risorse e i potenziali guasti dei container.

- Gruppo di contenitori daemon: prendete in considerazione l'utilizzo di un gruppo di contenitori daemon per processi in background o di monitoraggio che non richiedono la scalabilità rispetto al gruppo di contenitori primario. Ciò verifica che le attività essenziali in background vengano gestite in modo efficiente senza influire sui processi principali del server di gioco.

Passaggi dell'implementazione

- Progetta gruppi di contenitori con un contenitore principale per il server di gioco e l' GameLift agente e sidecar per la gestione delle dipendenze, con limiti specifici di memoria e CPU.
- Imposta limiti di risorse per ogni gruppo di contenitori per riservare le risorse necessarie e consentire al contempo un utilizzo controllato delle risorse per evitare contese.
- Utilizza un gruppo di contenitori daemon per attività in background o di monitoraggio, assicurandoti che funzionino in modo efficiente senza influire sui processi principali del server di gioco.

Calcolo e hardware

GAMEPERF04: Come si fa a impedire che le sessioni di gioco influiscano sui giocatori che utilizzano la stessa istanza del server di gioco?

Dopo l'avvio del server di gioco AWS, devi monitorarne le prestazioni per offrire un'esperienza di gioco di qualità indipendentemente dall'utilizzo delle risorse, dall'elaborazione sottostante o dalla saturazione.

Best practice

- GAMEPERF04-BP01 Monitora i processi del server di gioco per rilevare problemi
- GAMEPERF04-BP02 Metti alla prova le prestazioni del tuo server di gioco con scenari di gioco simulati e reali

GAMEPERF04-BP01 Monitora i processi del server di gioco per rilevare problemi

È possibile eseguire più processi del server di gioco per istanza per utilizzare in modo efficiente le risorse sulle istanze del server di gioco. In tal caso, progetta la tua architettura in modo che un

singolo processo del server di gioco che ospita una sessione di gioco non possa avere un impatto negativo su altre sessioni di gioco ospitate sulla stessa istanza. Usa le metriche per capire in che modo il posizionamento del gioco e il tipo di modalità di gioco possono influire sulle prestazioni delle istanze del server di gioco. Incorpora un mix di processi a basso carico (lobby, shop o tutorial per giocatore singolo) e ad alto carico (gameplay classificato, multigiocatore o con abilità elevate) per evitare di individuare l'istanza del server di gioco.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Monitora l'esperienza del giocatore attraverso metriche lato client e lato server raccogliendo dati telemetrici relativi al tempo di ping e al jitter, al calo dei frame, al tempo di risposta dell'API, agli errori e al completamento del loop di gioco con successo. Collega i timestamp di questi eventi ai problemi di assistenza ai giocatori e ai log del server per identificare i punti deboli nelle prestazioni. Strumenti come Dtrace, ftrace, uperf ed eBPF possono essere utilizzati per indagini e analisi approfondite delle prestazioni del sistema.

Implementa il monitoraggio delle risorse limitate disponibili per le istanze del server di gioco in modo da poter generare avvisi quando i singoli processi del server di gioco superano le soglie predeterminate di budget per le risorse. Quando le soglie vengono superate, potresti voler configurare il software del server di gioco in modo da scaricare i log del sistema e del server di gioco pertinenti su un dispositivo di archiviazione durevole, ad esempio una soluzione di registrazione centralizzata, in modo che gli ingegneri del server di gioco possano esaminare questo comportamento. Inoltre, l'istanza del server di gioco deve essere configurata per riportare le metriche di ciascuno dei processi del server di gioco in esecuzione sull'istanza in modo da poter monitorare questi singoli processi del server di gioco oltre alle metriche complessive per l'istanza del server di gioco.

Ad esempio, GameLift fornisce metriche per il monitoraggio delle sessioni di gioco, che possono essere integrate con metriche e log personalizzati specifici del gioco raccolti utilizzando CloudWatchAmazon Agent che puoi configurare sull'istanza del tuo server di gioco. Le tue metriche possono essere visualizzate CloudWatch o esportate in altri strumenti come Amazon Managed Grafana, che è integrato con Single Sign-On per semplificare l'accesso ai parametri da parte di utenti che potrebbero non avere accesso alla Console di gestione. Fai riferimento alle seguenti best practice per la gestione di log e metriche con Amazon GameLift, che fornisce anche supporto per la visualizzazione dei registri delle singole sessioni di gioco.

Passaggi dell'implementazione

- Esegui più processi del server di gioco per istanza e mescola modalità di gioco a basso e alto carico per evitare gli hot spotting e verificare un utilizzo equilibrato delle risorse.
- Monitora le metriche lato client e lato server come ping, jitter, cadute di frame e tempi di risposta delle API e mettile in correlazione con i log del server e i problemi segnalati dai giocatori per identificare i punti deboli.
- Configura il monitoraggio delle risorse per ogni processo del server di gioco, genera avvisi per le violazioni delle soglie e archivia i log in uno spazio di archiviazione durevole per l'analisi utilizzando strumenti come CloudWatch Amazon Managed Grafana.

GAMEPERF04-BP02 Metti alla prova le prestazioni del tuo server di gioco con scenari di gioco simulati e reali

Esegui test delle prestazioni e valuta vari scenari di gioco per determinare se il processo del server di gioco gestisce l'utilizzo di risorse fisse in modo appropriato, come memoria delle EC2 istanze, CPU e larghezza di banda di rete.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

La creazione di test di gioco simulati con bot in grado di rispecchiare percorsi e comportamenti di gioco comuni dei giocatori può determinare in che modo i processi del server di gioco gestiscono questo problema in diversi scenari di utilizzo. Ad esempio, puoi implementare una soluzione, come Distributed Load Testing, AWS che puoi personalizzare per eseguire simulazioni di client di gioco o build di client di gioco per generare scenari di gioco. Esegui test di gioco interni e utilizza i team di controllo qualità per testare le varie funzionalità del gioco in modo da poter acquisire la certezza che il gioco sia progettato per funzionare in modo ottimale. AWS Device Farmpuò essere utilizzato per eseguire test su dispositivi mobili e web per i giochi iOS, Android e browser su più tipi di dispositivi.

Passaggi dell'implementazione

- Esegui test delle prestazioni con bot che simulano i comportamenti comuni dei giocatori per valutare l'utilizzo delle risorse del server di gioco in diversi scenari.
- Utilizza soluzioni come Distributed Load Testing on AWS per personalizzare e simulare scenari di gioco per lo stress test.

- Esegui playtest interni e utilizza strumenti come i test di gioco AWS Device Farm per dispositivi mobili e browser su vari dispositivi.

Calcola la selezione

GAMEPERF05: Come scegli la soluzione di calcolo appropriata per il tuo gioco?

Le prestazioni di calcolo variano a seconda delle dimensioni e delle famiglie di istanze. È vantaggioso utilizzare più opzioni di elaborazione provenienti da pool di capacità separati. Sviluppa una strategia di composizione della flotta che dia la preferenza alle prestazioni ma includa una diversità sufficiente per evitare errori di capacità insufficienti.

Best practice

- GAMEPERF05-BP01 Valuta le tue prestazioni di gioco su più tipi di elaborazione
- GAMEPERF05-BP02 Sposta le attività di calcolo in flussi di lavoro asincroni non-latency-sensitive

GAMEPERF05-BP01 Valuta le tue prestazioni di gioco su più tipi di elaborazione

Per i carichi di lavoro dei server di gioco, non esiste un approccio univoco per identificare la soluzione di elaborazione ottimale per ospitare il server di gioco. Una strategia comune per il benchmarking dei server di gioco consiste nell'iniziare con istanze EC2 «c» ottimizzate per il computer, poiché questa famiglia di istanze offre prestazioni elevate per carichi di lavoro ad alta intensità di calcolo. In alternativa, se il gioco richiede una notevole quantità di memoria per implementare funzionalità specifiche, le istanze ottimizzate per la memoria potrebbero essere le più adatte.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Se il tuo carico di lavoro utilizza notevoli risorse di rete, valuta la possibilità di implementare istanze ottimizzate per la rete, come in genere indicato con una «n» nel nome dell'istanza, ed evita istanze burstable di tipo «t», poiché una volta esauriti i crediti, le prestazioni diminuiranno. I giochi sono sensibili alla latenza e alla perdita di pacchetti, quindi si consiglia di utilizzare una rete EC2 avanzata per migliorare le prestazioni di rete dei server di gioco. La rete avanzata utilizza la I/O virtualizzazione

a root singolo (SR-IOV) per fornire funzionalità di rete ad alte prestazioni sui tipi di istanze supportati. SR-IOV è un metodo di virtualizzazione dei dispositivi che offre I/O prestazioni più elevate e un minore utilizzo della CPU rispetto alle tradizionali interfacce di rete virtualizzate. Le reti avanzate forniscono infatti una larghezza di banda più alta, prestazioni PPS (pacchetti al secondo) superiori e latenze tra istanze significativamente più basse. La rete avanzata con Elastic Network Adapter è disponibile per i tipi di EC2 istanze più recenti ed è importante aggiornarla regolarmente per beneficiare dei miglioramenti delle prestazioni offerti dalle istanze più recenti e dai miglioramenti apportati all'hypervisor Nitro.AWS

Se il tuo gioco funziona in modo simile su più tipi di EC2 istanze, dovresti prendere in considerazione l'utilizzo di più tipi di istanze per ospitare i tuoi server di gioco. Monitora le prestazioni nel tempo ed esegui ulteriori ottimizzazioni dopo aver ospitato un numero sufficiente di sessioni di gioco di produzione da poter identificare le tendenze delle prestazioni. Ricorda che i tuoi requisiti di elaborazione possono cambiare man mano che aggiungi nuove funzionalità al gioco che richiedono una diversa allocazione delle risorse. Puoi configurare i gruppi di EC2 Auto Scaling per utilizzare più tipi di istanze oppure puoi utilizzare gruppi di Auto Scaling separati per ospitare istanze di server di gioco che eseguono tipi di istanze separati, il che potrebbe semplificare la gestione della correlazione e dell'aggregazione delle metriche.

Valuta le prestazioni del tuo gioco su diversi tipi di processori, ad esempio istanze basate su Intel, istanze basate su AMD e istanze Graviton basate su ARM. Unreal Engine 5.1.1 o versioni successive possono compilare server di gioco per Graviton e migliorare il rapporto prezzo/prestazioni del gioco. Esegui test di scansione e saturazione a varie dimensioni all'interno di ciascuna famiglia per determinare il punto ottimale in cui utilizzo e prestazioni sono coerenti.

Dovresti anche confrontare l'impatto sulle prestazioni del tuo gioco quando è ospitato utilizzando contenitori e funzioni Lambda. Per i casi d'uso in cui non sono richiesti processi di server di gioco di lunga durata, come i giochi asincroni e per i servizi di backend di gioco, prendi in considerazione l'utilizzo di un'architettura serverless con Lambda che può semplificare la gestione e le operazioni per i team addetti alle operazioni di gioco, oltre a consentire di distribuire più rapidamente il gioco a livello globale a molti. Regioni AWS Per le best practice serverless, consulta Serverless Applications Lens - Well-Architected Framework.

Passaggi dell’implementazione

- Effettua il benchmark dei server di gioco su istanze «c» ottimizzate per il computer per carichi di lavoro che richiedono un uso intensivo della CPU, istanze ottimizzate per la memoria per attività che richiedono molta memoria e istanze «n» ottimizzate per la rete per un throughput di rete elevato.

- Utilizza una rete avanzata con Elastic Network Adapter (ENA) sulle istanze supportate per migliorare le prestazioni di rete, ridurre la latenza e aumentare la velocità di elaborazione dei pacchetti.
- Valuta e testa diversi tipi di istanze, processori (Intel, AMD, Graviton) e opzioni di hosting in container o Lambda, adattando le soluzioni di elaborazione all'evolversi delle funzionalità di gioco.

Per ulteriori informazioni, consulta Scegli la strategia di elaborazione giusta per i tuoi server di gioco globali.

GAMEPERF05-BP02 Sposta le attività di calcolo in flussi di lavoro asincroni non-latency-sensitive

Quando ottimizzi le prestazioni del gioco, è importante tenere presente che solo alcune delle interazioni tra il client e il backend del gioco devono essere eseguite in modo sincrono. Dovresti considerare ogni funzionalità dal punto di vista dell'esperienza del giocatore e determinare se determinate interazioni richiedono comunicazioni sincrone, che sono bloccanti e richiedono molte risorse, o se tali funzionalità possono essere implementate in modo asincrono. Quando implementate le chiamate di rete, utilizzate un approccio asincrono e non bloccante. Inoltre, il backend di gioco dovrebbe essere configurato per eseguire il lavoro in modo efficiente, scaricando le attività in coda e dando priorità alle risposte rapide ai client, ove possibile.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Ad esempio, l'aggiornamento di una classifica al termine di una sessione di gioco può essere implementato in modo asincrono in modo asincrono in modo che il client non debba attendere il completamento dell'aggiornamento della classifica. Implementalo invece in modo asincrono sul client di gioco e valuta la possibilità di progettare il tuo servizio di backend per inserire questo tipo di operazioni in coda, come Amazon SQS. Con questa architettura, configura il backend in modo che accetti la richiesta, inseriscilo in SQS per archiviare in modo duraturo i messaggi per l'elaborazione asincrona e rispondi tempestivamente al client. Una volta completato l'aggiornamento della classifica, il backend può inviare un aggiornamento al client di gioco in modo che la visualizzazione della classifica da parte del giocatore venga aggiornata.

In alternativa, il giocatore può semplicemente visitare la schermata della classifica del gioco per recuperare i dati più recenti, che può inviare una richiesta web al backend per recuperare i dati più recenti dalla cache.

Passaggi dell'implementazione

- Determina se le interazioni client-backend richiedono una comunicazione sincrona; implementa approcci asincroni e non bloccanti, ove possibile, per ottimizzare l'utilizzo delle risorse.
- Usa Amazon SQS per ridurre il carico di attività non critiche come gli aggiornamenti della classifica.
- Consenti al client di recuperare i dati aggiornati in modo asincrono, ad esempio recuperando i dati più recenti della classifica su richiesta o tramite aggiornamenti in background.

Resources

- Comprensione della messaggistica asincrona per i microservizi
- Lambda - Utilizzo delle integrazioni di servizi e dell'elaborazione asincrona

Gestione dei dati

GAMEPERF06: Come gestisci e analizzi in modo efficiente i registri dei server di gioco e archivia diversi tipi di dati di gioco per prestazioni ottimali?

I giochi possono contenere dati dei giocatori, registri di gioco e registri del server che devono essere disaccoppiati l'uno dall'altro, ove possibile. La centralizzazione dell'inserimento dei log e della gestione del ciclo di vita può portare vantaggi ai team di gioco, in quanto fornisce informazioni dettagliate su ciò che accade nel gioco e sul server.

Best practice

- GAMEPERF06-BP01 Centralizza la raccolta e l'archiviazione dei log
- GAMEPERF06-BP02 Categorizza e archivia i dati di gioco in base ai modelli di accesso
- GAMEPERF06-BP03 Abilita la formattazione e il raggruppamento efficienti dei log
- GAMEPERF06-BP04 Implementare politiche di rotazione e conservazione dei log
- GAMEPERF06-BP05 Usa strumenti di monitoraggio e visualizzazione

GAMEPERF06-BP01 Centralizza la raccolta e l'archiviazione dei log

Implementa una soluzione centralizzata di raccolta e archiviazione dei log per raccogliere i log dalle istanze dei server di gioco e. GameLift

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Utilizza servizi come Amazon CloudWatch Logs per raccogliere, monitorare e archiviare i dati di registro dai server e dalle GameLift istanze di gioco. CloudWatch Logs offre una soluzione scalabile e completamente gestita per la gestione dei log, che facilita l'archiviazione e il recupero efficienti dei dati di registro senza influire sulle prestazioni dei server di gioco. Se utilizzi l'agente CloudWatch Logs, prendi in considerazione i vari tipi di installazione e le opzioni di configurazione, come la dimensione del batch e la durata del buffer, per ridurre al minimo l'impatto sul server di gioco. Considera le istanze del server di gioco effimere e riduci la dipendenza dalla registrazione localizzata, ove possibile. Stabilisci una politica centralizzata per l'implementazione delle migliori pratiche di registrazione.

Passaggi dell'implementazione

- Usa Amazon CloudWatch Logs per raccogliere, monitorare e archiviare i dati di registro dalle istanze dei server di gioco e per facilitare la gestione centralizzata e GameLift scalabile dei log.

GAMEPERF06-BP02 Categorizza e archivia i dati di gioco in base ai modelli di accesso

Categorizza i dati di gioco in diversi tipi in base ai modelli di accesso e ai requisiti di archiviazione.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Le categorie più comuni includono i dati dei giocatori, i salvataggi di gioco, l'archiviazione mondiale persistente e i dati di analisi.

Passaggi dell'implementazione

Utilizza soluzioni di archiviazione appropriate per ogni tipo di dati per ottimizzare le prestazioni e l'efficienza in termini di costi:

- Dati dei giocatori: usa Amazon DynamoDB, un database NoSQL veloce e scalabile, per archiviare profili, preferenze e dati di progressione dei giocatori. L'accesso a bassa latenza e le funzionalità di scalabilità automatica di DynamoDB forniscono un recupero e un aggiornamento efficienti dei dati dei giocatori.
- Giochi salvati: usa Amazon S3 per archiviare i salvataggi di gioco e i checkpoint. S3 offre durabilità e scalabilità elevate per archiviare grandi quantità di dati di gioco salvati. Prendi in considerazione l'utilizzo di S3 Transfer Acceleration o Amazon CloudFront per caricare e scaricare più rapidamente i salvataggi dei giochi.
- Storage mondiale persistente: per giochi con stati mondiali persistenti o dati di gioco condivisi, prendi in considerazione l'utilizzo di Amazon DynamoDB, Amazon o ElastiCache Amazon MemoryDB. ElastiCache e MemoryDB forniscono un archivio chiave-valore in memoria mentre DynamoDB è un database NoSQL supportato da SSD. Questi servizi forniscono un accesso rapido ai dati archiviati, riducendo il tempo impiegato dal processo del server di gioco per salvare lo stato del gioco, il che migliora le prestazioni complessive del processo.
- Dati di analisi: usa Amazon Managed Streaming for Apache Kafka o Kinesis Data Streams per importare flussi di dati dai tuoi produttori di dati di gioco. Amazon Managed Service per Apache Flink può essere utilizzato per la trasformazione e l'analisi in tempo reale e inviato ad Amazon Data Firehose per l'elaborazione e la distribuzione in data lake, magazzini e servizi di analisi di backend. La guida per Game Analytics Pipeline on AWS illustra come i servizi interagiscono per fornire analisi quasi in tempo reale e in batch.

GAMEPERF06-BP03 Abilita la formattazione e il raggruppamento efficienti dei log

Configura i processi del server di gioco per generare log in un formato strutturato e analizzabile, come JSON.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Implementa tecniche di log batching per ridurre al minimo la frequenza dei trasferimenti di dati di log dai server di gioco all'archiviazione centralizzata dei log. Il batch dei log riduce il sovraccarico della rete e migliora le prestazioni dei server di gioco. Utilizza i log dettagliati o a livello di debug come eccezione e non come impostazione predefinita, poiché possono comportare un calo delle prestazioni e dei costi che dovrebbe essere evitato quando possibile.

GAMEPERF06-BP04 Implementare politiche di rotazione e conservazione dei log

Stabilisci politiche di rotazione e conservazione dei log per gestire la crescita dei dati di registro e ottimizzare l'utilizzo dello storage.

Livello di rischio associato se questa best practice non fosse adottata: basso

Guida all’implementazione

Configura i tuoi server di gioco per ruotare automaticamente i log in base alle dimensioni o agli intervalli di tempo. Definisci politiche di conservazione dei log in Amazon CloudWatch Logs per archiviare o eliminare automaticamente i dati di registro più vecchi che non sono più necessari per l'analisi attiva o la risoluzione dei problemi.

GAMEPERF06-BP05 Usa strumenti di monitoraggio e visualizzazione

Utilizza strumenti di monitoraggio e visualizzazione per ottenere informazioni dettagliate sulle prestazioni del server di gioco e identificare opportunità di ottimizzazione.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Usa Amazon CloudWatch per monitorare i parametri chiave e configurare allarmi per notifiche proattive. Utilizza strumenti come Amazon Managed Service for Prometheus e Amazon Managed Grafana per raccogliere, interrogare e visualizzare le metriche dei tuoi server di gioco e della tua infrastruttura. Crea dashboard informativi per monitorare le prestazioni, identificare i punti deboli ed effettuare ottimizzazioni basate sui dati.

Reti e distribuzione di contenuti

GAMEPERF07: Come progetti il tuo servizio di matchmaking per ottimizzare le prestazioni?

L'abilità dei giocatori, la qualità del provider di servizi Internet (ISP) e la distribuzione della popolazione di giocatori sono importanti da considerare come una dimensione di ottimizzazione delle prestazioni. Le sessioni di gioco possono essere collocate su server posizionati strategicamente per livellare il campo di gioco e ospitare un gioco equo.

Best practice

- GAMEPERF07-BP01 Definisci le soglie di latenza di rete per il tuo gioco
- GAMEPERF07-BP02 Esegui un servizio di matchmaking separato per ogni modalità di gioco e regione di hosting del gioco
- GAMEPERF07-BP03 Monitora regolarmente le prestazioni del matchmaking
- GAMEPERF07-BP04 Monitora regolarmente le prestazioni di rete
- GAMEPERF07-BP05 Utilizza la tecnologia di accelerazione della rete per migliorare le prestazioni su Internet

GAMEPERF07-BP01 Definisci le soglie di latenza di rete per il tuo gioco

Quando sviluppi un gioco multiplayer, verifica che la tua infrastruttura di gioco non introduca latenze non necessarie per i giocatori. Se il tuo gioco è sensibile alla latenza di rete, allora dovresti impostare delle soglie di latenza nella tua logica di matchmaking per dare priorità all'inserimento dei giocatori in sessioni di server di gioco ospitate in posizioni di server di gioco vicine o Regioni AWS che soddisfano il tuo obiettivo di un'esperienza di gioco ideale.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

In molti giochi sensibili alla latenza è comune utilizzare i client di gioco per eseguire il ping di ciascuna posizione dell'infrastruttura del gioco per raccogliere dati sulle prestazioni come latenza di rete, jitter e perdita di pacchetti e riportare questi dati al backend di raccolta delle metriche in modo che possano essere analizzati. Quando abbini i giocatori alle sessioni di gioco, puoi configurare il gioco per incorporare la latenza di rete percepita dal client di gioco nell'infrastruttura del tuo server di gioco come uno degli input utilizzati nella logica del servizio di matchmaking.

GAMEPERF07-BP02 Esegui un servizio di matchmaking separato per ogni modalità di gioco e regione di hosting del gioco

Se il tuo gioco offre ai giocatori più modalità di gioco tra cui scegliere, dovresti separare i sistemi di matchmaking per ognuna di esse in modo da poter ottimizzare le prestazioni di ciascuna modalità di gioco in base ai suoi requisiti specifici e ridurre la contesa di risorse. Ogni modalità di gioco avrà probabilmente requisiti unici in termini di latenza accettabile, dimensione della partita e altre logiche di matchmaking personalizzate specifiche del gioco. Probabilmente attireranno anche diversi tipi di giocatori. Esegui il servizio di matchmaking di ciascuna modalità di gioco come distribuzione

software separata in modo da poter testare le prestazioni e utilizzare le modalità di gioco in modo indipendente.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Ad esempio, potresti eseguirle come funzioni Lambda separate per ogni modalità di gioco oppure utilizzarle come distribuzioni di servizi separate basate su contenitori.

Distribuisci i tuoi servizi di matchmaking in più regioni vicine alle posizioni dei tuoi server di gioco. Il traffico dei giocatori seguirà molti percorsi, quindi è importante che il servizio di matchmaking mantenga un profilo di up-to-date latenza su più percorsi per migliorare l'efficienza del posizionamento delle sessioni di gioco ISPs a bassa latenza. GameLift FlexMatch fornisce ulteriori indicazioni per la selezione delle regioni per i matchmaker e include la possibilità di integrare i matchmaker con code di sessioni di gioco multiregionali.

GAMEPERF07-BP03 Monitora regolarmente le prestazioni del matchmaking

Uno dei modi più evidenti per ottimizzare le prestazioni di un gioco per i giocatori consiste nel ridurre il tempo di attesa prima di poter accedere a una sessione di gioco. I lunghi tempi di attesa possono far perdere interesse ai giocatori e portare al logoramento, quindi è importante tenerne conto quando si progetta la soluzione di matchmaking.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Quando stai progettando la configurazione di matchmaking per il tuo gioco, crea regole che determinino le condizioni da applicare per formare una partita. Dovresti considerare l'impatto che queste regole avranno sulle prestazioni del sistema, in particolare i tempi di attesa per i giocatori. Prima di apportare modifiche all'implementazione del matchmaking, come l'aggiunta di nuove condizioni o filtri di matchmaking, testatelo prima o valutate la possibilità di applicare questa modifica gradualmente a un piccolo campione di giocatori, come canarino, oppure A/B provate a raccogliere dati sulle prestazioni prima di introdurre questa modifica all'intera popolazione di giocatori.

Configura il tuo servizio di matchmaking per generare log dettagliati per comprendere le condizioni o le regole applicate a ciascuna richiesta di matchmaking. Ciò aiuta nella revisione e nell'adeguamento dell'implementazione del matchmaking, se necessario.

Per esempio, Amazon GameLift FlexMatch fornisce un servizio di matchmaking completamente gestito che può essere utilizzato come servizio autonomo con il tuo hosting di server di gioco o utilizzato con server di gioco ospitati su Amazon. GameLift FlexMatch può generare notifiche di eventi per Amazon EventBridge, vedi Configurare notifiche di FlexMatch eventi. Usa Amazon Simple Notification Service (Amazon SNS) per ricevere dati di matchmaking in formato JSON, che ti consente di elaborare e archiviare automaticamente queste informazioni per l'analisi e migliorare le prestazioni del matchmaking.

Imposta metriche per tenere traccia del tempo impiegato dal tuo servizio di matchmaking per trovare una sessione di gioco adatta ai giocatori. Rivedi regolarmente le metriche sulla durata del matchmaking e confronta questi tempi con il comportamento dei giocatori e il sentimento della community. Usa questi dati per sviluppare soglie adeguate per i timeout del matchmaking che possono essere incluse nella configurazione delle regole di matchmaking.

Ad esempio, Amazon GameLift FlexMatch fornisce supporto per la definizione dei timeout delle richieste di matchmaking e per la creazione di regole di matchmaking che possono consentire un allentamento dei requisiti nel tempo. Questa funzionalità ti consente di creare un matchmaking adattabile per semplificare la creazione di partite e inserire i giocatori in sessioni di gioco quando le partite sono difficili da trovare.

GAMEPERF07-BP04 Monitora regolarmente le prestazioni di rete

Per i giochi competitivi, è importante avere un'esperienza di gioco coerente.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Una partita che dura 50 ms per una base di giocatori più ampia è più equa e divertente di una partita in cui un giocatore ha un ping di 10 ms e un altro di 70 ms. Le modifiche al routing dell'ISP possono influire su una parte della popolazione di giocatori, e il tuo sistema di matchmaking dovrà adattarsi. Amazon CloudWatch Network Monitoring ti aiuta a determinare se il problema riguarda il tuo gioco o il provider Internet del giocatore.

Passaggi dell’implementazione

- Usa Amazon Cloudwatch Network Monitoring per monitorare le prestazioni di rete e identificare i problemi di routing.

- Usa i log di flusso VPC per identificare modelli di traffico anomali o pacchetti persi, il che può indicare congestione della rete, problemi con l'ISP o configurazioni errate che influiscono sulla latenza del giocatore.

GAMEPERF07-BP05 Utilizza la tecnologia di accelerazione della rete per migliorare le prestazioni su Internet

Oltre a collocare fisicamente l'infrastruttura di gioco sensibile alla latenza più vicina ai giocatori, puoi anche migliorare l'esperienza dei giocatori ottimizzando le prestazioni di rete per il gioco. AWS utilizza il protocollo BGP per influenzare il routing di Internet in modo da utilizzare il percorso più veloce verso la nostra rete di confine dai provider di servizi Internet. Se gestisci la tua rete e hai bisogno di maggiore controllo e visibilità sul comportamento di routing e sulla pubblicità BGP, puoi utilizzare il Peering privato o indirizzare il traffico da Internet Direct Connect al gioco in esecuzione. AWS

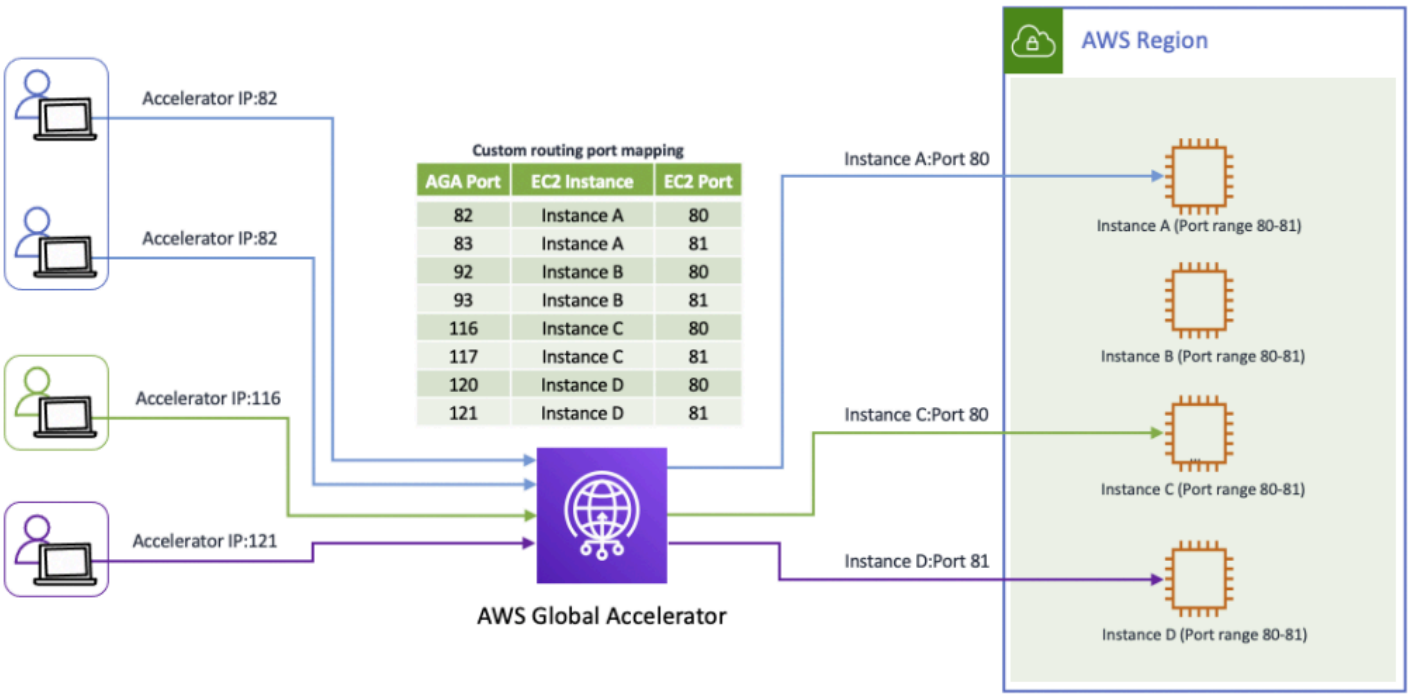
Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

Prendi in considerazione la seguente architettura di riferimento per supportare prestazioni e tempi di risposta Internet migliorati.

Prestazioni di rete migliorate per i giochi con Global Accelerator

Per una soluzione completamente gestita per il routing di rete, AWS Global Accelerator migliora le prestazioni di rete dell'applicazione utilizzando la rete AWS globale, che può essere utilizzata per accelerare il traffico di gioco, la chat vocale e il traffico di messaggistica in tempo reale, nonché altre applicazioni sensibili alla latenza, fornendo al contempo un failover rapido ai server di gioco. Gli acceleratori di routing personalizzati Global Accelerator possono essere integrati con il tuo servizio di matchmaking per fornire un routing deterministico di più giocatori alla stessa sessione di gioco utilizzando indirizzi IP e porte statici anycast.



I tuoi team di sviluppo di giochi possono essere distribuiti in tutto il mondo e richiedere un accesso efficiente a contenuti o risorse condivisi. Per migliorare le prestazioni dei contenuti condivisi archiviati nei bucket Amazon S3, puoi configurare la replica bidirezionale dei dati tra le regioni utilizzando S3 Cross-Region Replication in modo che gli utenti possano accedere ai dati dai bucket più vicini a loro. Per semplificare questo modello di accesso, utilizza S3 Multi- Region Access Point che accelera le richieste a S3 sulla rete globale utilizzando Global Accelerator.

Per ulteriori informazioni, consulta Improving the Player Experience by Leveraging AWS Global Accelerator e Amazon GameLift FleetiQ.

Passaggi dell'implementazione

- Usa AWS Global Accelerator per migliorare le prestazioni di rete per il traffico di gioco, la chat vocale e la messaggistica in tempo reale, facilitando al contempo il failover rapido sui server di gioco.
- Configura gli acceleratori di routing personalizzati di Global Accelerator da integrare con il tuo servizio di matchmaking, abilitando il routing deterministico dei giocatori alle sessioni di gioco utilizzando anycast statico. IPs
- Abilita S3 Cross-Region Replication per replicare i contenuti condivisi tra le regioni per i team di sviluppo di giochi distribuiti.

- Usa i punti di accesso multiregionali S3 per accelerare l'accesso ai dati S3 sulla rete globale per gli utenti distribuiti a livello globale. AWS

Processo e cultura

GAMEPERF08: Come allinea la barra delle prestazioni del tuo gioco alle aspettative di giocatori e sviluppatori?

Conoscere il proprio giocatore e sviluppatore è uno degli aspetti più importanti per migliorare l'efficienza delle prestazioni. Offrire un gioco performante con costi operativi ridotti è uno dei modi migliori per dimostrare a giocatori e sviluppatori che tieni alla loro esperienza e che puoi differenziare gioco e studio.

Best practice

- GAMEPERF08-BP01 Informa e includi il giocatore nel tuo processo
- GAMEPERF08-BP02 Allinea la selezione della soluzione alle competenze e all'esperienza del team di progettazione

GAMEPERF08-BP01 Informa e includi il giocatore nel tuo processo

Fornisci un'opzione per visualizzare nel gioco metriche come latenza, frame per secondo e pacchetti eliminati. Evidenzia i problemi dell'infrastruttura e i tempi di inattività dovuti alla manutenzione attraverso comunicazioni rivolte ai giocatori, come le pagine di stato. Celebrate le nuove ambientazioni di gioco con le comunicazioni dei giocatori, compresi i blog degli sviluppatori, e stabilite le aspettative relative ai miglioramenti previsti per l'esperienza dei giocatori.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Includi il giocatore

Fornisci una semplice procedura di invio diagnostico che raccolga i file pertinenti e li alleghi a un ticket di assistenza giocatori dal tuo client di gioco. Attiva un forum di supporto in cui i giocatori possano aiutarsi a vicenda e contribuire a migliorare l'esperienza di gioco

Valuta i compromessi rispetto alle aspettative dei giocatori

I giocatori potrebbero non accorgersi dello spostamento dei sistemi di backend per ridurre i costi, ma spostare i server di gioco può modificare il tempo di ping. Sii coerente e leale nei confronti dei giocatori motivando l'espansione e la riduzione delle sedi di hosting dei giochi.

Le comunità e le aree geografiche di giocatori avranno caratteristiche proprie che potrebbero influire sulle aspettative del gioco. Ad esempio, la Corea del Sud ha alcune delle connessioni Internet più veloci del pianeta e l'aspettativa di gioco è una latenza a una cifra, che favorisce un gioco altamente competitivo. Il gameplay casual su dispositivi mobili crea un profilo prestazionale e un modello di utilizzo diversi rispetto alle sessioni di gioco su console e PC.

L'accesso e la lobby fanno parte dell'esperienza e dovrebbero essere reattivi, anche se il server è offline per manutenzione. La pianificazione dei raid notturni o il tempo libero nella lobby fanno parte dell'esperienza del giocatore ed è importante tenerne conto quando si scelgono aree di interesse per l'efficienza delle prestazioni. I giocatori possono lasciare il client di gioco aperto per mesi, a volte possono semplicemente accedere di tanto in tanto per leggere le note sulla patch. Un gioco Live Ops deve tenere a mente l'intera esperienza del giocatore come parte del processo e della cultura ingegneristica.

Passaggi dell’implementazione

- Fornisci parametri di gioco come latenza, FPS e perdita di pacchetti e comunica i problemi dell'infrastruttura e i programmi di manutenzione tramite pagine di stato e aggiornamenti rivolti ai giocatori.
- Implementa una funzionalità di dump diagnostico e invio nel client di gioco e crea un forum di supporto per promuovere la risoluzione dei problemi e il miglioramento guidati dalla community.
- Personalizza le ottimizzazioni delle prestazioni in base alle aspettative della community di giocatori, ad esempio con bassa latenza per le regioni competitive o esperienze reattive per giocatori occasionali e con sessioni prolungate. login/lobby
- Progetta flussi di lavoro Live Ops per tenere conto dell'intera esperienza del giocatore, dal gameplay attivo al comportamento inattivo del cliente, facilitando il coinvolgimento senza interruzioni.

GAMEPERF08-BP02 Allinea la selezione della soluzione alle competenze e all'esperienza del team di progettazione

Valuta le capacità e l'esperienza del tuo team nella gestione e ottimizzazione delle prestazioni dei server di gioco al momento di scegliere l'opzione di hosting che preferisci. Le soluzioni self-hosted come EC2 i container richiedono una maggiore conoscenza della gestione dell'infrastruttura, dell'ottimizzazione delle prestazioni e della scalabilità. Se il tuo team non dispone di queste competenze, un servizio gestito GameLift potrebbe essere più adatto, in quanto elimina molte complessità e consente al team di concentrarsi sulle ottimizzazioni specifiche del gioco.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Valutando questi fattori ed eseguendo test prestazionali tra diverse opzioni di hosting, puoi selezionare la soluzione più appropriata che soddisfi i requisiti specifici del tuo gioco ottimizzando al contempo l'efficienza delle prestazioni.

Resources

Scopri di più sulle nostre best practice relative all'efficienza delle prestazioni.

Documenti correlati:

- AWS Architecture Center
- Pilastro dell'efficienza delle prestazioni: AWS Well-Architected Framework
- Confronto dei modelli di storage in locale con i servizi di storage AWS
- Instance Store: storage temporaneo a blocchi per le EC2 istanze
- Raccogli metriche, log e tracce utilizzando l'agente CloudWatch
- CloudWatchAgente
- Come posso attivare e configurare una rete avanzata sulle mie EC2 istanze?
- Migliorare l'esperienza dei giocatori sfruttando AWS Global Accelerator e Amazon FleetiQ GameLift
- Blog sulla tecnologia di Riot Games: scalabilità e test di carico per Valorant
- Giochi online iperscalabili con una soluzione ibrida AWS
- Metodo di saturazione ed errori di utilizzo (USE)

- Istanze Amazon EC2 Spot
- Prestazioni su larga scala con Amazon ElastiCache
- Strategie di memorizzazione nella cache del database che utilizzano Redis
- Opzioni di connettività di Amazon Virtual Private Cloud
- Modelli di progettazione basati sulle best practice: ottimizzazione delle prestazioni di Amazon S3

Benchmark correlati:

- Effettua il benchmark dei volumi Amazon EBS
- Larghezza di banda di rete delle istanze Amazon EC@

Strumenti correlati:

- Unreal Engine: test e ottimizzazione dei contenuti
- Unity Profiler
- Sistema di qualità Open 3D Engine (O3DE)
- Monitoraggio dei GameLift server Amazon
- Toolkit GameLift di test Amazon

Video correlati:

- AWS re:Invent 2019: [RIPETI 2] Amazon EC2 Foundations (-R2) CMP211
- AWS re:Invent 2019: Potenziamento di EC2 Amazon di nuova generazione: approfondimenti sul sistema Nitro (03-R2) CMP3
- Guida introduttiva ad Amazon GameLift FleetiQ AWS — Online Tech Talks
- Zach Blitz di Riot Games parla dell'uso per migliorare il gioco AWS
- AWS re:Invent 2023 - AWS Graviton: il miglior rapporto prezzo/prestazioni per i tuoi carichi di lavoro () AWS CMP334

Formazione correlata:

- Hosting di server di gioco su AWS
- Utilizzo di Amazon GameLiftFleet IQ per server di gioco

- Guida introduttiva a AWS for Games — Parte I
- Hosting di server di gioco su AWS
- AWS re:Invent 2023 — AWS Graviton: il miglior rapporto prezzo/prestazioni per i tuoi carichi di lavoro () AWS CMP334

Ottimizzazione dei costi

Il pilastro dell'ottimizzazione dei costi include il processo continuo di perfezionamento e miglioramento di un sistema durante l'intero ciclo di vita. Questo processo va dalla progettazione iniziale del primo proof of concept al funzionamento continuo dei carichi di lavoro di produzione. Adottando le pratiche descritte in questo paper, è possibile creare e gestire sistemi attenti ai costi che consentono di raggiungere i risultati aziendali desiderati al prezzo più basso. L'implementazione di queste pratiche di ottimizzazione dei costi può consentire alla vostra azienda di massimizzare il valore del vostro investimento nel cloud.

I giochi sono progetti creativi unici che devono competere per l'attenzione e il tempo di gioco dei giocatori. Prima del lancio, gli sviluppatori di giochi spesso non hanno una chiara comprensione di quanto sarà popolare o duraturo il loro gioco. A seconda della strategia di monetizzazione del gioco, delle priorità aziendali e della fase del ciclo di vita, gli sviluppatori dovranno fare dei compromessi nel valutare le decisioni di ottimizzazione dei costi.

Ad esempio, durante la fase di pre-lancio di un nuovo gioco tanto atteso, l'attenzione si concentra in genere sullo sviluppo delle funzionalità e sulle prestazioni. speed-to-market La priorità è verificare che l'infrastruttura sia scalabile per soddisfare i picchi di domanda dei giocatori. Al contrario, se un gioco non ha successo o lo sviluppo rallenta, l'attenzione può spostarsi sulla riduzione del più possibile dei costi per continuare a far funzionare il gioco per i giocatori esistenti.

Molti sviluppatori di giochi gestiscono anche più giochi contemporaneamente, il che richiede ulteriori considerazioni. Risorse come l'infrastruttura, il software e lo staff possono essere condivise tra più partite dal vivo, permettendo di compensare le perdite di un gioco con i profitti di un altro. In questo scenario, concentrarsi sull'ottimizzazione dei costi può migliorare le finanze dell'intero portafoglio di giochi.

Dati i modelli di business unici, la scalabilità e l'imprevedibilità dei giochi, le seguenti domande chiave possono guidare le decisioni di ottimizzazione dei costi:

- Come posso misurare il costo dell'infrastruttura per giocatore, sistema e funzionalità di gioco?
- Qual è il giusto equilibrio tra ottimizzazione dei costi ed esperienza del giocatore per l'attuale fase del ciclo di vita del mio gioco?
- Come posso massimizzare il ritorno sull'investimento utilizzando il modello di prezzo giusto per le mie risorse? AWS

L'applicazione di queste best practice e la formulazione delle domande giuste possono aiutare gli sviluppatori di giochi a creare e gestire sistemi attenti ai costi che consentono di ottenere risultati aziendali riducendo al minimo i costi.

Aree di interesse

- Principi di progettazione
- Implementazione della gestione finanziaria del cloud
- Comprensione delle spese e dell'utilizzo
- Risorse convenienti in termini di costo
- Costi per il trasferimento dati
- Gestione della domanda e delle risorse di offerta
- Ottimizzazione nel tempo
- Resources

Principi di progettazione

Oltre ai principi di progettazione del pilastro dell'ottimizzazione dei costi del Well-Architected Framework, i seguenti principi di progettazione ottimizzano i costi di esecuzione del carico di lavoro di gioco nel cloud.

- Misura il costo dell'infrastruttura per giocatore, sistema e funzionalità di gioco: comprendi e monitora i costi di infrastruttura necessari per esperienze e funzionalità specifiche dei giocatori su tutti i sistemi di gioco. In questo modo è possibile identificare le aree dell'architettura che potrebbero richiedere l'ottimizzazione dei costi.
- Valuta il compromesso tra l'ottimizzazione dei costi e l'esperienza del giocatore: valuta in quale fase si trova il gioco per determinare l'obiettivo giusto: esperienza del giocatore o ottimizzazione dei costi. In genere, una volta che un gioco raggiunge la massa critica e la popolazione di giocatori si è stabilizzata, è il momento di concentrarsi sull'ottimizzazione dei costi operativi. La chiave è bilanciare l'offerta di un'ottima esperienza di gioco con la gestione dell'infrastruttura di gioco nel modo più conveniente. L'implementazione di questi principi di progettazione massimizza il ritorno sugli investimenti nei giochi.

Implementazione della gestione finanziaria del cloud

Non esistono best practice di Cloud Financial Management specifiche per Games Lens. Fai riferimento al Well-Architected Framework Cost Optimization Pillar per indicazioni sulla gestione finanziaria del cloud.

Comprensione delle spese e dell'utilizzo

GAMECOST01: Come misurate il costo dei vostri ambienti di gioco?

Comprendi il costo per giocatore, la funzionalità di gioco e l'ambiente in modo da poter gestire e prevedere la spesa man mano che il numero di giocatori cambia nel tempo e le funzionalità vengono aggiunte e migliorate. Prendi in considerazione le seguenti best practice per gestire i costi dei diversi ambienti di gioco.

Best practice

- GAMECOST01-BP01 Implementa l'attribuzione del costo per giocatore, funzionalità di gioco e ambiente
- GAMECOST01-BP02 Scopri le opportunità di ottimizzazione

GAMECOST01-BP01 Implementa l'attribuzione del costo per giocatore, funzionalità di gioco e ambiente

L'attribuzione dei costi per i server di gioco è in genere più semplice da eseguire rispetto ai servizi di backend di gioco, poiché un server di gioco è generalmente ottimizzato per ospitare un numero specifico di giocatori simultanei per istanza, che può essere ammortizzato in base al costo di esecuzione dell'istanza.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Per i servizi di backend di gioco, si consiglia di suddividere i componenti del gioco in funzionalità distinte che possono essere gestite come risorse logiche o fisiche separate per semplificare l'analisi dei costi.

Ad esempio, sebbene possa sembrare semplice implementare una singola applicazione monolitica per ospitare i servizi di backend di gioco, questo modello rende difficile ricavare il costo totale per giocatore e funzionalità di gioco nel tempo man mano che si aggiungono altre funzionalità perché i costi di elaborazione, rete e archiviazione delle risorse sono condivisi tra le funzionalità. Prendi in considerazione l'adozione di un'architettura serverless per i tuoi servizi di backend di gioco con servizi come Amazon API Gateway AWS Lambda e/o AWS Fargate per l'elaborazione, Amazon SQS e Amazon SNS per la messaggistica, Amazon S3 per lo storage di oggetti e Amazon DynamoDB per lo storage di database. Questi servizi sono solo alcuni esempi di prodotti che offrono prezzi basati sull'utilizzo e basati principalmente sul volume delle richieste, in modo da poter visualizzare i costi con granularità. Le singole risorse come le funzioni Lambda, i servizi Fargate, le tabelle DynamoDB e i bucket S3 possono essere associate ai tag di allocazione dei costi in modo da attribuire ai costi di questi servizi nomi di funzionalità di gioco che semplificano la comprensione dei costi di ciascuno dei servizi.

Si consiglia inoltre di gestire separatamente ciascuno degli ambienti di sviluppo dei giochi in modo da poter attribuire i costi ai diversi ambienti. In genere, gli sviluppatori di giochi gestiranno ambienti separati per lo sviluppo, il test, la messa in scena e la produzione, come descritto nel pilastro operativo di questo settore dei giochi. Ogni ambiente presenta in genere requisiti di scalabilità, prestazioni e utilizzo diversi e può essere gestito da team separati. Per controllare i costi, organizza questi ambienti in modo da poter monitorare e attribuire correttamente i costi di ciascun ambiente.

Per ulteriori informazioni, consulta la seguente documentazione:

- Creazione di un gioco multigiocatore senza server scalabile
- Server di sessione di gioco autonomi con un backend basato WebSockets
- Server di sessione di gioco autonomi con un backend senza server

Passaggi dell’implementazione

- Separa i servizi di backend di gioco in funzionalità distinte utilizzando architetture serverless o containerizzate come Amazon AWS Lambda API Gateway AWS Fargate e abilita l'attribuzione granulare dei costi per funzionalità.
- Applica i tag di allocazione dei costi alle singole risorse (ad esempio, funzioni Lambda, tabelle DynamoDB e bucket S3) per associare i costi a funzionalità di gioco specifiche per una migliore analisi dei costi.
- Gestisci ambienti separati per lo sviluppo, il test, l'allestimento e la produzione, organizzando e monitorando i relativi costi in modo indipendente per allinearli ai requisiti di scalabilità e utilizzo.

GAMECOST01-BP02 Scopri le opportunità di ottimizzazione

Gli sviluppatori e gli editori di giochi possono utilizzare AWS FinOps pratiche per ottimizzare i costi del cloud e ottenere una migliore visibilità sulla spesa per il cloud. In questo modo, i produttori di giochi possono allineare il costo medio richiesto per la manutenzione dell'infrastruttura per i giocatori ai risultati finanziari ottenuti dal gioco.

Livello di rischio associato se questa best practice non fosse adottata: basso

Guida all’implementazione

AWS offre una guida alla soluzione pronta all'uso per Cloud Financial Management per gestire e ottimizzare le spese per i servizi cloud. Questa funzionalità include visibilità granulare e analisi di costi e utilizzo per supportare il processo decisionale su argomenti come i dashboard di spesa, l'ottimizzazione, i limiti di spesa, il riaddebito e il rilevamento e la risposta delle anomalie. La guida alla soluzione per Cloud Financial Management include funzionalità di budget e previsione, che offrono un'architettura definita e ottimizzata in termini di costi per i carichi di lavoro in modo da poter selezionare il modello di prezzo giusto e attribuire i costi delle risorse pertinenti ai team. Ciò attiva tecniche di tracciamento, notifica e ottimizzazione dei costi in tutto l'ambiente e le risorse. È possibile gestire centralmente le informazioni sulle spese e consentire alle parti interessate critiche di accedervi secondo necessità per una visibilità mirata e per supportare il processo decisionale.

Un altro FinOps strumento chiave è il Cost Optimization Hub, che offre una visione centralizzata dei consigli e delle opportunità di ottimizzazione dei costi a livello Regioni AWS nazionale, in modo che possiate ottenere il massimo dalla vostra AWS spesa. Account AWS Puoi utilizzare Cost Optimization Hub per identificare, filtrare e aggregare i consigli per l'ottimizzazione dei AWS costi a livello aziendale Account AWS . Regioni AWS Fornisce consigli sul ridimensionamento dei diritti delle risorse, sull'eliminazione delle risorse inattive, su Savings Plans e sulle istanze riservate. Con un'unica dashboard, eviti di dover utilizzare più AWS prodotti per identificare le opportunità di ottimizzazione dei costi.

Se i team di gioco utilizzano Account AWS la funzionalità condivisa MyApplications in Console di gestione AWS Home, può essere utilizzata per visualizzare i costi delle risorse applicative per i singoli carichi di lavoro. Questa visualizzazione granulare ti consente di identificare le tendenze specifiche dei costi all'interno della tua infrastruttura di gioco, consentendoti di prendere decisioni informate sull'allocazione e l'ottimizzazione delle risorse.

Inoltre, la revisione regolare dei dati di fatturazione e gestione dei costi con AWS Data Exports rivela opportunità nascoste di risparmio sui costi. Questo rapporto dettagliato fornisce un'analisi completa

della spesa per il cloud, consentendoti di identificare le aree di spesa eccessiva, le risorse non utilizzate e le opportunità di sfruttare servizi o modelli di prezzo più convenienti.

Adottando FinOps i principi e sfruttando gli strumenti forniti da AWS, gli sviluppatori e gli editori di giochi possono utilizzare nel modo più efficiente le proprie risorse cloud, migliorando in ultima analisi i profitti e liberando fondi per lo sviluppo e l'innovazione dei giochi.

Passaggi dell'implementazione

- Utilizza gli strumenti di gestione Cloud AWS finanziaria per una visibilità granulare e dettagliata, dashboard di spesa, rilevamento delle anomalie e attribuzione dei costi per ottimizzare e tenere traccia delle spese relative al cloud in modo efficace.
- Utilizza il Cost Optimization Hub per centralizzare i consigli sul rightsizing, sul Savings Plans e sulle istanze riservate in tutte le regioni. Account AWS
- Esamina regolarmente i dati di AWS fatturazione utilizzando Data Exports e così MyApplication via AWS per analizzare i costi specifici del carico di lavoro, scoprire opportunità di risparmio e ottimizzare l'allocazione delle risorse.

Risorse convenienti in termini di costo

GAMECOST02: Come stai scegliendo la soluzione di elaborazione giusta per i tuoi server di gioco?

Uno degli aspetti più singolari del carico di lavoro di gioco, rispetto ad altri tipi di carichi di lavoro, è il server di gioco. Il server di gioco è fondamentale per l'esperienza del giocatore, poiché i giocatori si connettono ad esso dal proprio client di gioco per giocare una sessione di gioco.

Il server di gioco è anche uno dei principali fattori di costo per la gestione di un gioco multiplayer. Pertanto, è importante ottimizzare il modo in cui utilizzi l'infrastruttura di calcolo per i tuoi server di gioco per ridurre i costi.

Best practice

- GAMECOST02-BP01 Ottimizza il costo del trasferimento dei dati su Internet
- GAMECOST02-BP02 Ottimizza il numero di sessioni di gioco ospitate su ciascuna istanza del server di gioco per ottimizzare i costi

- GAMECOST02-BP03 Seleziona l'opzione di calcolo appropriata per ridurre i costi

GAMECOST02-BP01 Ottimizza il costo del trasferimento dei dati su Internet

Sebbene siano AWS principalmente addebitati i costi per il trasferimento dei dati in uscita (in uscita) dalle AWS risorse dell'utente a Internet, le società produttrici di giochi possono sostenere costi elevati legati al trasferimento dei dati tramite AWS Direct Connect i sistemi di bilanciamento del carico AWS Gateway, che possono addebitare sia i dati in entrata (in ingresso) che quelli in uscita. Implementa soluzioni che riducano il costo complessivo del trasferimento dei dati dal AWS backend del gioco ai giocatori, concentrandoti sulla riduzione al minimo dei costi di uscita dalle tue AWS risorse e sulla valutazione delle opzioni per gestire le tariffe di ingresso e uscita tramite i servizi di AWS connettività.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Usa Amazon CloudFront per ridurre i costi della distribuzione di contenuti e delle applicazioni web rivolte al pubblico.

I contenuti e le risorse di gioco archiviati nel cloud vengono in genere archiviati in Amazon S3 e consegnati al client di gioco direttamente da S3 o dai server Web ospitati su Amazon EC2 che recuperano i contenuti da Amazon S3 e li distribuiscono ai client. Per ridurre i costi di trasferimento dei dati legati al download di contenuti, prendi CloudFront in considerazione l'idea di utilizzare Amazon davanti al tuo spazio di archiviazione cloud per distribuire contenuti agli utenti.

L'utilizzo consente di CloudFront ridurre i costi di trasferimento dei dati, poiché la distribuzione dei contenuti è più economica CloudFront points-of-presence rispetto a quella direttamente dalle regioni e CloudFront non addebita costi di recupero dell'origine per le origini AWS basate, come Amazon EC2 e Amazon S3. Se i contenuti sono statici e non vengono modificati spesso, è possibile memorizzarli CloudFront nella cache più vicino agli utenti finali, in modo da ridurre ulteriormente i costi.

CloudFront migliora inoltre l'efficienza in termini di costi delle applicazioni e dei servizi Web frontali rivolti al pubblico, anche se non viene utilizzata la memorizzazione nella cache, poiché il costo del trasferimento dei dati tra server e client può essere ridotto instradando il traffico attraverso la rete. AWS

Amazon CloudWatch può essere utilizzato per monitorare CloudFront l'utilizzo di Amazon. Per i casi d'uso in cui utilizzi più reti di distribuzione di contenuti (CDNs), Amazon CloudFront Origin Shield

può fornire un ulteriore livello di caching per consolidare e ridurre il numero di richieste di origine da diversi provider.

Per comprendere il traffico della rete di gioco, puoi abilitare VPC Flow Logs e Amazon CloudWatch Internet Monitor per avere end-to-end visibilità sulle connessioni backend dei giocatori o dei giochi. Questo approccio può identificare le cause degli elevati costi di trasferimento dei dati ed eseguire modifiche architetturali per ottimizzare la spesa per il trasferimento dei dati.

Passaggi dell'implementazione

- Usa Amazon di fronte ad Amazon S3 o CloudFront a origini di contenuti EC2 basate su Amazon per ridurre i costi di trasferimento dei dati sfruttando la distribuzione a basso costo da CloudFront points-of-presence e rimuovendo i costi di recupero dell'origine.
- Abilita VPC Flow Logs e Amazon CloudWatch Internet Monitor per analizzare il traffico di rete e identificare le modifiche architetturali per ottimizzare i costi di trasferimento dei dati.
- Implementa CloudFront Origin Shield per consolidare e ridurre le richieste di origine quando ne usi più di una CDNs per una maggiore efficienza dei costi.

Per ulteriori best practice per la distribuzione dei contenuti, consulta il white paper Content Delivery for Games.

GAMECOST02-BP02 Ottimizza il numero di sessioni di gioco ospitate su ciascuna istanza del server di gioco per ottimizzare i costi

Ottimizza il numero di sessioni di gioco ospitate per istanza del server per ottenere un migliore utilizzo dell'elaborazione e ridurre i costi dell'infrastruttura di elaborazione.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Per ottimizzare i costi, gli sviluppatori di giochi dovrebbero massimizzare il numero di sessioni di gioco ospitate sullo stesso server fisico o virtuale, noto anche come densità di imballaggio dei server di gioco. Ciò si ottiene aumentando il numero di processi dei server di gioco che possono essere ospitati simultaneamente su un'istanza.

Un singolo processo del server di gioco in genere non dovrebbe richiedere l'uso di tutte le risorse disponibili sull' EC2 istanza. Questo è uno dei modi più importanti per ridurre i costi di elaborazione

di un gioco e richiede l'uso di software in grado di generare e gestire più processi server sull' EC2 istanza su porte separate.

Ad esempio, Amazon GameLift ha una quota sul numero massimo di processi del server di gioco per istanza, che dovresti sforzarti di utilizzare in modo da ridurre i costi di hosting. Per ulteriori informazioni, consulta la sezione Endpoints e quote di Amazon GameLift Servers per i dettagli sulla quota attuale per il numero massimo di processi dei server di gioco per istanza.

In alternativa all'implementazione dei processi dei server di gioco su macchine virtuali come EC2 le istanze, sta diventando sempre più popolare tra gli sviluppatori di giochi utilizzare i propri server di gioco come applicazioni basate su contenitori utilizzando soluzioni di orchestrazione dei container. Gli sviluppatori di giochi possono utilizzare Amazon Elastic Container Service (Amazon ECS) o Guidance for Game Server Hosting utilizzando Agones e Open Match su Amazon EKS. Un'altra opzione è Game Server Hosting on AWS Fargate, un motore di elaborazione serverless che funziona sia con ECS che con EKS, che ti consente di concentrarti sul gioco senza dover gestire l'infrastruttura sottostante.

Le soluzioni container offrono funzionalità di pianificazione dei lavori in grado di trovare automaticamente un'istanza di container disponibile nel cluster per ospitare il container del server di gioco in base ai requisiti di risorse e ad altre logiche di posizionamento specificate dall'utente. Tuttavia, è importante considerare in che modo gestirete il ridimensionamento e il comportamento di posizionamento dei giocatori in modo da non interrompere le sessioni attive dei giocatori.

Passaggi dell’implementazione

- Aumenta la densità di impacchettamento eseguendo più processi su server di gioco per EC2 esempio utilizzando porte e software di gestione dei processi separati.
- Usa Amazon GameLift o soluzioni container come ECS, EKS o AWS Fargate per gestire i processi dei server di gioco in modo efficiente e ridurre i costi dell'infrastruttura.
- Monitora continuamente l'utilizzo delle risorse per affinare la densità di imballaggio e mantenere l'efficienza dei costi senza compromettere l'esperienza del giocatore.

GAMECOST02-BP03 Seleziona l'opzione di calcolo appropriata per ridurre i costi

Esegui test delle prestazioni del software del tuo server di gioco su una varietà di tipi di istanze e opzioni di calcolo per determinare quale opzione è più conveniente per il tuo gioco.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Oltre a utilizzare in modo efficiente i tipi di EC2 istanze giusti per il tuo carico di lavoro, valuta quale delle opzioni di calcolo disponibili è più adatta ai tuoi obiettivi di ottimizzazione dei costi. Sono disponibili diverse opzioni di prezzo, tra cui On-Demand Instances, Spot Instances, Reserved Instances e Savings Plans.

I Savings Plans (SPs) offrono sconti per l'elaborazione assumendo impegni di utilizzo e sono ideali per gli scenari in cui non è possibile prevedere l'utilizzo previsto per un periodo di 1 o 3 anni. Offrono sconti come le istanze riservate con la flessibilità di applicarli a diverse regioni, famiglia di istanze, sistema operativo e locazione. Possono essere applicati anche a AWS Fargate chi può fungere da server di gioco per giochi occasionali o a AWS Lambda chi viene utilizzato come ottima opzione per giochi a turni che non richiedono server di gioco. Per ulteriori informazioni, consulta Creazione di un gioco multigiocatore senza server scalabile.

I Savings Plans vengono introdotti durante il lancio del gioco per ridurre i costi dei carichi di lavoro dei server di gioco che contribuiscono alla spesa in EC2 istanze quando il gioco viene rilasciato al pubblico. I Savings Plans possono essere introdotti anche dopo il lancio quando il team addetto alle operazioni di gioco ha una migliore comprensione del traffico dei giocatori dopo che il gioco è rimasto in produzione per un lungo periodo.

Poiché i Savings Plans offrono flessibilità regionale, sono particolarmente ideali per ottimizzare la spesa dei server di gioco per giochi con un utilizzo imprevedibile in tutte le aree geografiche.

Ad esempio, se il tuo modello di utilizzo giornaliero da parte dei giocatori richiede almeno 20 server per supportare la tua base di giocatori, ma periodicamente richiede fino a 40 server, prendi in considerazione l'acquisto di impegni del Savings Plan per coprire la linea di base dei 20 server, poiché tale domanda di utilizzo è prevedibile e coerente e si tradurrà in un utilizzo massimo dell'impegno di utilizzo acquistato.

Massimizza l'utilizzo di Savings Plans e ampliali con altre opzioni di acquisto che offrono maggiore flessibilità per picchi di utilizzo imprevedibili dei server di gioco, come le istanze on-demand e Spot per ottenere risparmi ottimali.

Le istanze Spot sono ideali per gestire server di gioco perché offrono i maggiori sconti di calcolo, non richiedono impegni di utilizzo e offrono flessibilità per tipi di carichi di lavoro imprevedibili e con picchi di lavoro. Tuttavia, le istanze Spot possono essere interrotte, quindi sono più adatte per carichi

di lavoro su server di gioco con sessioni di gioco di breve durata o situazioni in cui la tolleranza all'interruzione è maggiore.

Per ulteriori informazioni sulle linee guida per l'esecuzione di server di gioco utilizzando Kubernetes su Amazon EKS con istanze EC2 Spot, consulta Come eseguire giochi multigiocatore di massa con Spot EC2 utilizzando Aurora Serverless.

Utilizza le istanze Amazon EC2 Spot per determinare i pool con il minor rischio di interruzione che offriranno il massimo risparmio rispetto alle tariffe on demand.

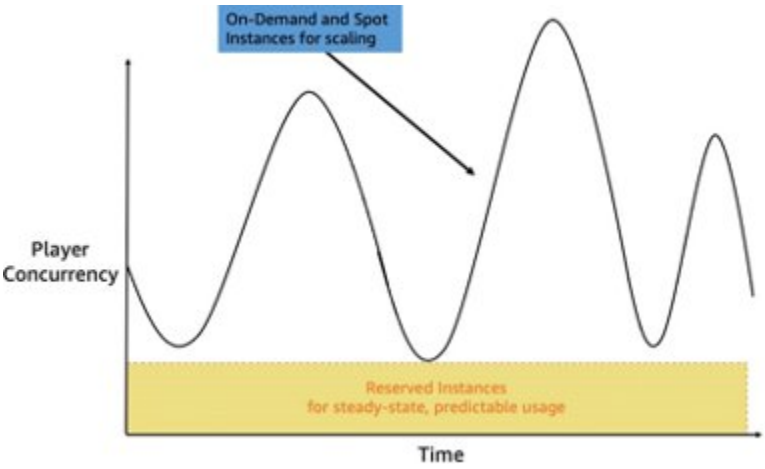
Quando si utilizza Spot, si consiglia inoltre di eseguire carichi di lavoro sui server di gioco su più tipi di EC2 istanze e zone di disponibilità in Regione AWS modo da diversificare l'utilizzo della capacità e ridurre il rischio di interruzione.

Valuta la possibilità di utilizzare le istanze Spot in combinazione con le istanze on demand per ridurre al minimo l'impatto di potenziali interruzioni nelle sessioni di gioco attive e utilizza una strategia di allocazione ottimizzata della capacità per ridurre ulteriormente il rischio di interruzione.

Per ulteriori best practice, consulta le Best practice per Amazon EC2 Spot. Il ribilanciamento della capacità in Auto Scaling per sostituire le istanze Spot a rischio può essere utilizzato per monitorare in modo proattivo e aggiungere capacità aggiuntiva quando le istanze Spot sono maggiormente a rischio di interruzione.

Amazon GameLift FleetiQ si integra con le istanze Spot per ottimizzare l'uso di istanze Spot a basso costo riducendo al contempo il rischio di interruzioni. Se stai ospitando il gioco utilizzando GameLift, consulta la documentazione per la scelta delle risorse informatiche. GameLift Per ulteriori informazioni, consulta Scegliere le risorse di calcolo per una flotta gestita.

Il diagramma seguente fornisce un esempio per illustrare l'uso di più opzioni di calcolo dei prezzi per i carichi di lavoro dei server di gioco:



Hosting di server di gioco con diverse opzioni di EC2 prezzo

Nel diagramma, la concorrenza tra i giocatori oscilla nel tempo, il che rende difficile la gestione dell'utilizzo e l'ottimizzazione dei costi. Per ovviare a questa fluttuazione, prendi in considerazione l'adozione di una combinazione di diverse opzioni di calcolo dei prezzi, utilizzando Savings Plans EC2 per soddisfare le esigenze dei requisiti minimi di utilizzo e affidandoti alle istanze EC2 On-Demand e EC2 Spot per soddisfare le esigenze dei giocatori.

Passaggi dell'implementazione

- Utilizza Savings Plans per un utilizzo di base prevedibile, combinandoli con istanze Spot e On-Demand per la flessibilità e l'ottimizzazione dei costi durante i picchi di utilizzo.
- Usa le istanze Spot per server di gioco con sessioni di breve durata o maggiore tolleranza alle interruzioni, diversificando i tipi di istanze e le zone di disponibilità per ridurre al minimo i rischi.
- Implementa strumenti come EC2 Spot Instances Advisor, Capacity Rebalancing e GameLift FleetiQ per ottimizzare l'utilizzo delle istanze Spot e gestire in modo proattivo le interruzioni.

Costi per il trasferimento dati

GAMECOST03: Come stai ottimizzando i costi di trasferimento dei dati per la tua infrastruttura di gioco?

I giochi possono trasferire una quantità significativa di dati su Internet tra i dispositivi client di gioco dei giocatori e l'infrastruttura di gioco per fornire l'esperienza di gioco, nonché tra i componenti dell'infrastruttura di gioco.

Ad esempio, il trasferimento dei dati avviene quando i giocatori scaricano gli aggiornamenti dei contenuti di gioco sui propri client di gioco, salvano lo stato di avanzamento del gioco nel cloud, partecipano a sessioni di gioco multiplayer in tempo reale con i propri amici e quando l'infrastruttura di gioco trasferisce dati tra regioni e zone di disponibilità. È importante capire dove avviene il trasferimento dei dati durante il carico di lavoro di gioco per ottimizzare le scelte di architettura e ridurre i costi di trasferimento dei dati.

Per ottimizzare i costi di trasferimento dei dati per il carico di lavoro di gioco, prendi in considerazione le seguenti best practice:

Best practice

- GAMECOST03-BP01 Scegli il tipo di storage appropriato per i contenuti generati dagli utenti per ridurre i costi
- GAMECOST03-BP02 Ottimizza i database per i backend di gioco

GAMECOST03-BP01 Scegli il tipo di storage appropriato per i contenuti generati dagli utenti per ridurre i costi

Ogni tipo di dati generato e archiviato nel gioco presenta caratteristiche uniche che è necessario prendere in considerazione per determinare la soluzione di archiviazione più adatta al carico di lavoro.

Livello di rischio associato se questa best practice non fosse adottata: basso

Guida all'implementazione

Usa Amazon S3 Object Lifecycle Management per archiviare i dati degli oggetti nella classe di storage più conveniente. Amazon S3 offre diverse classi di storage e la gestione del ciclo di vita degli oggetti per semplificare la configurazione di policy semplici e dettagliate per la transizione automatica dei dati tra i livelli di storage per ridurre i costi. Invece di archiviare semplicemente i dati nella classe di storage S3 standard per impostazione predefinita, prendi in considerazione la possibilità di impostare una configurazione del ciclo di vita per trasferire automaticamente i dati tra i livelli nel tempo, oppure utilizza la classe di storage S3 Intelligent-Tiering per modelli di accesso sconosciuti o in evoluzione.

In alternativa, S3 Intelligent-Tiering può trasferire in modo economico e automatico i dati tra i livelli ed è consigliata come classe di storage predefinita poiché offre l'ottimizzazione dei costi senza la necessità di impostare manualmente le politiche del ciclo di vita ed è ora la scelta migliore per oggetti piccoli e di breve durata. Per ulteriori informazioni, consulta Amazon S3 Intelligent-Tiering — Ottimizzazione dei costi migliorata per oggetti di breve durata e di piccole dimensioni.

I casi d'uso più comuni di Amazon S3 includono lo storage di risorse di gioco, contenuti statici, log di gioco, archiviazione di data lake e backup. Per i casi d'uso in cui sono necessari file system, come il collegamento di file system condivisi alle workstation durante lo sviluppo, prendi in considerazione l'utilizzo di Amazon Elastic File System (Amazon EFS), che offre diverse classi di storage e cresce e si riduce automaticamente man mano che aggiungi e rimuovi file senza dover gestire l'infrastruttura.

Amazon S3 One Zone -IA è un'opzione di archiviazione ideale per dati transitori relativi a sessioni di gioco, matchmaking o altre informazioni effimere che possono essere ricreate secondo necessità. Questo tipo di dati di gioco non richiede ridondanza tra più zone di disponibilità (). AZs Questa classe di archiviazione a basso costo è ideale per registrare le azioni dei giocatori, gli eventi di gioco e altri dati di telemetria utilizzati per l'analisi o il debug.

Il principale vantaggio di ottimizzazione dei costi dell'utilizzo di S3 Express One Zone per tali dati di gioco è il notevole risparmio sui costi rispetto alla classe di archiviazione S3 standard, con una riduzione fino al 20% dei costi di archiviazione. Questo può essere particolarmente vantaggioso per i giochi con grandi volumi di dati che non richiedono lo stesso livello di durabilità e disponibilità dei dati delle applicazioni mission-critical. Sfruttando S3 One Zone, gli sviluppatori e gli editori di giochi possono ottimizzare i costi di archiviazione sul cloud senza compromettere l'esperienza complessiva del giocatore.

Passaggi dell’implementazione

- Configura le politiche del ciclo di vita di Amazon S3 per trasferire i dati tra classi di storage o usa S3 Intelligent-Tiering come impostazione predefinita per l'ottimizzazione automatica dei costi con modelli di accesso in evoluzione.
- Usa S3 One Zone-Infrequent Access per i dati delle sessioni di gioco temporanee, come i record di telemetria e matchmaking, per ridurre i costi di archiviazione fino al 20% mantenendo al contempo una disponibilità sufficiente.
- Per esigenze di file system condivise durante lo sviluppo, usa Amazon EFS per semplificare la gestione dello storage con capacità elastica e più classi di storage.

GAMECOST03-BP02 Ottimizza i database per i backend di gioco

I giochi fanno molto affidamento sui database per archiviare un'ampia gamma di dati critici, dai profili e gli inventari dei giocatori alle microtransazioni e alle metriche di progressione all'interno del gioco. I database svolgono anche un ruolo cruciale nella gestione degli aspetti sociali dei giochi, come la creazione e il mantenimento di gruppi di giocatori, feste e l'applicazione delle politiche di moderazione. Man mano che la base di giocatori di un gioco cresce, i costi associati ai database aumenteranno inevitabilmente per soddisfare le crescenti richieste di dati e utilizzo.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Per i backend di gioco in esecuzione su Amazon Aurora, è possibile utilizzare diverse strategie di ottimizzazione dei costi. Una raccomandazione fondamentale è quella di ridimensionare automaticamente le repliche di lettura in base ai modelli di utilizzo, aumentando o diminuendo dinamicamente il numero di repliche per gestire le fluttuazioni del traffico. Ciò significa che paghi per le risorse di cui hai veramente bisogno. Un'altra tattica di ottimizzazione consiste nel sostituire le repliche di lettura utilizzate per l'analisi dei giochi con l'esportazione di snapshot DB in Amazon S3, poiché il servizio di storage S3 è generalmente più conveniente rispetto alle istanze di database Aurora fornite. Per ulteriori informazioni, consulta Esportazione di dati di snapshot DB in Amazon S3 per Amazon RDS.

L'esplorazione dell'uso di istanze DB riservate per Amazon Aurora per le istanze di database principali e la transizione alla configurazione Aurora Serverless possono anche portare a notevoli risparmi sui costi a lungo termine fornendo maggiore flessibilità e controllo granulare sull'utilizzo delle risorse.

Analogamente, per i backend di gioco che utilizzano Amazon DynamoDB, l'utilizzo della modalità di capacità on-demand di DynamoDB può essere una scelta efficace, soprattutto per carichi di lavoro nuovi o imprevedibili, in quanto consente di pagare solo per le risorse consumate senza la necessità di fornire eccessivamente. Man mano che i modelli di traffico di gioco diventano più stabili e prevedibili nel tempo, puoi passare alla modalità di capacità fornita da DynamoDB, che può offrire risparmi sui costi attraverso una migliore pianificazione della capacità. L'attivazione dell'auto-scaling sulle tabelle DynamoDB è un'altra ottimizzazione chiave, che consente al servizio di regolare dinamicamente la capacità fornita in base alle fluttuazioni del traffico. Testa la struttura dei dati del gioco in un ambiente di sviluppo prima del lancio per trovare e rimuovere gli indici secondari locali () e gli indici secondari globali () non necessari. LSIs GSIs Ciò può portare a notevoli risparmi sui costi per l'archiviazione e le operazioni dei dati di gioco. La rimozione di operazioni di scansione inefficienti dal codice di backend di gioco a favore di query più mirate, l'acquisto di capacità riservata di Amazon DynamoDB e l'utilizzo di DynamoDB Streams con AWS Lambda trigger per elaborare gli eventi di backend di gioco possono ottimizzare ulteriormente i costi di DynamoDB. Per ulteriori informazioni, consulta Best practice per l'interrogazione e la scansione dei dati in DynamoDB.

Implementando queste strategie di ottimizzazione dei costi sia per Amazon Aurora che per DynamoDB, gli sviluppatori e gli editori di giochi possono ridurre in modo significativo la spesa per i database di backend di gioco.

Passaggi dell’implementazione

- Usa l'auto-scaling delle repliche di lettura di Aurora e le esportazioni di snapshot DB su Amazon S3 per una gestione efficiente in termini di costi delle fluttuazioni del traffico e delle esigenze di analisi.
- Ottimizza i costi di DynamoDB iniziando con la capacità su richiesta per nuovi carichi di lavoro, passando alla capacità fornita con auto-scaling per traffico prevedibile e rimuovendo i e inutilizzati. LSIs GSIs
- Evita operazioni di scansione inefficienti a favore di query mirate, utilizza le istanze riservate o la capacità riservata e utilizza DynamoDB Streams per l'elaborazione degli eventi. AWS Lambda

Gestione della domanda e delle risorse di offerta

Non esistono best practice specifiche per la gestione della domanda e dell'offerta in materia di risorse per Games Lens.

Per ulteriori informazioni sulla gestione della domanda e sulla fornitura di risorse, vedere Cost Optimization Pillar - Well-Architected AWS Framework.

Ottimizzazione nel tempo

Non esistono best practice di ottimizzazione nel tempo specifiche per Games Lens.

Per ulteriori informazioni sull'ottimizzazione dei costi nel tempo, vedere Cost Optimization Pillar - Well-Architected AWS Framework.

Resources

Fai riferimento alle seguenti risorse per saperne di più sulle migliori pratiche per l'ottimizzazione dei costi:

Documenti correlati:

- Come posso ridurre i costi di trasferimento dati per il mio gateway NAT in Amazon VPC?
- Presentazione dell'adattatore Amazon GameLift FleetiQ per Agones
- Come posso trovare i principali contributori al traffico del gateway NAT nel mio Amazon VPC?
- Scegli la strategia di elaborazione giusta per i tuoi server di gioco globali
- AWS Well-Architected Labs: risorse convenienti

- Il plug-in Amazon VPC CNI aumenta i limiti di pod per nodo
- Le migliori pratiche di architettura per l'ottimizzazione dei costi
- Riduzione dei tempi di attesa dei giocatori e dimensionamento corretto dell'allocazione di calcolo con Amazon SageMaker AI RL e Amazon EKS
- AWS Compute Optimizer
- Electronic Arts ottimizza i costi e le operazioni di storage utilizzando Amazon S3 Intelligent-Tiering e Amazon Glacier
- Evita pratiche di licenza ostili migrando i carichi di lavoro Windows su Linux
- Overview of Data Transfer Costs for Common Architectures
- AWS e Kubecost collaborano per fornire il monitoraggio dei costi ai clienti EKS
- Gettare le basi: configurare l'ambiente per l'ottimizzazione dei costi
- Panoramica delle istanze Amazon EC2 Spot

Sostenibilità

Il pilastro della sostenibilità fornisce principi di progettazione, linee guida operative, best practice e piani di miglioramento per contribuire a raggiungere gli obiettivi di sostenibilità per i AWS carichi di lavoro.

È possibile trovare linee guida sull'implementazione nel white paper Sustainability Pillar - AWS Well-Architected Framework.

Aree di interesse

- Principi di progettazione
- Selezione della regione
- Allineamento alla domanda
- Software e architettura
- Gestione dei dati
- Hardware e servizi
- Resources

Principi di progettazione

La sostenibilità dell'industria dei giochi si sta evolvendo a ritmi diversi nelle diverse regioni del mondo. Con l'energia sostenibile in vaste aree del Nord America e i mandati di sostenibilità nell'UE e nel Regno Unito, gli architetti hanno diversi approcci per raggiungere carichi di lavoro più ecologici nel prossimo futuro. Il pilastro della sostenibilità Well-Architected può essere utilizzato per raggiungere queste misure per un'ampia varietà di carichi di lavoro.

Questa sezione dell'obiettivo descrive diverse best practice che possono essere utilizzate per i carichi di lavoro legati ai giochi.

- Seleziona il tipo di archiviazione appropriato per i dati utente dei giochi.
- Fai attenzione alle politiche del ciclo di vita dei dati che deduplicano i dati ed eliminano i dati non necessari dai tuoi carichi di lavoro.
- Sii selettivo riguardo al corretto dimensionamento della distribuzione delle risorse di elaborazione.
- Usa serverless per processi brevi e transazionali.

Selezione della regione

Non esistono best practice per la selezione delle regioni specifiche per Games Lens. Per maggiori dettagli, consulta Sustainability Pillar - AWS Well-Architected Framework.

Allineamento alla domanda

Non vi è alcun allineamento alla richiesta di buone pratiche specifiche per Games Lens. Per maggiori dettagli, consulta Sustainability Pillar - AWS Well-Architected Framework.

Software e architettura

Non esistono best practice relative al software e all'architettura specifiche per Games Lens. Per maggiori dettagli, consulta Sustainability Pillar - AWS Well-Architected Framework.

Gestione dei dati

GAMESUS01: Come gestisci i dati utente e di gioco nel tuo sistema di gioco?

Sviluppa una strategia del ciclo di vita dei dati che ottimizzi l'archiviazione e la pertinenza dei dati conservando solo i dati storici critici.

Best practice

- GAMESUS01-BP01 Utilizza tecnologie di archiviazione che si adattano ai modelli adattati ai contenuti degli utenti, alle informazioni sugli abbonati e agli acquisti in-game
- GAMESUS01-BP02 Utilizza le politiche del ciclo di vita o la scadenza TTL per eliminare giochi, dati utente, file di registro o risorse obsolete non necessari.

GAMESUS01-BP01 Utilizza tecnologie di archiviazione che si adattano ai modelli adattati ai contenuti degli utenti, alle informazioni sugli abbonati e agli acquisti in-game

È necessario classificare i dati per tipo, esigenza di conservazione e frequenza di accesso. Ciò ti consente di selezionare la soluzione di archiviazione più ottimizzata per la miriade di tipi di dati

prodotti dai tuoi servizi di gioco o di backend. I dati in rapida evoluzione devono essere archiviati in servizi di database chiave-valore o in memoria. I dati transazionali devono essere archiviati in servizi di database relazionali. I file di grandi dimensioni, le risorse di gioco o i contenuti generati dagli utenti devono essere archiviati nei servizi di archiviazione degli oggetti.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all'implementazione

I giochi producono e utilizzano un'ampia varietà di tipi di dati che richiedono soluzioni di archiviazione ottimizzate per frequenza di accesso, latenza e costi. I dati archiviati devono essere classificati utilizzando tag per differenziare i dati che possono essere rimossi o che devono essere archiviati a lungo termine.

I seguenti servizi funzionano bene per una varietà di casi d'uso dei giochi:

Amazon Aurora (compatibile con MySQL e PostgreSQL) offre alta disponibilità, bassa latenza e scalabilità automatica, il che lo rende una scelta eccellente per gestire grandi quantità di dati transazionali, come la gestione e l'autenticazione degli account dei giocatori, le economie di gioco, le classife e le classiifiche dei giocatori, la persistenza dello stato del gioco, la gestione di eventi e campagne e l'implementazione multiregionale e ad alta disponibilità.

Amazon DynamoDB è un database NoSQL completamente gestito noto per la sua bassa latenza, il throughput elevato e la scalabilità perfetta, che lo rendono ideale per la gestione in tempo reale dei dati dei giocatori, la gestione delle sessioni, l'inventario, l'economia di gioco, lo stato delle partite multiplayer in tempo reale, il matchmaking, la registrazione degli eventi e la scalabilità per un pubblico globale.

Amazon DocumentDB (compatibile con MongoDB) fornisce un servizio di database orientato ai documenti scalabile e a bassa latenza, perfetto per archiviare dati flessibili e semistrutturati, come sistema di inventario, profili e personalizzazioni dei giocatori, mondi di gioco e contenuti generati proceduralmente, interazioni sociali e con i giocatori, analisi e monitoraggio del comportamento e metadati e configurazioni di gioco.

Amazon ElastiCache supporta il caching in memoria con Redis o Memcached, offrendo un accesso rapido ai dati e tempi di risposta ridotti, il che è fondamentale per i giochi multiplayer in tempo reale in cui velocità e prestazioni sono essenziali per un'esperienza utente fluida. ElastiCache viene utilizzato nei videogiochi per classificazioni in tempo reale, gestione delle sessioni, memorizzazione nella cache dei metadati di gioco, chat e messaggistica in-game, matchmaking, analisi e telemetria in tempo reale e scalabilità per eventi ad alto traffico.

Amazon Simple Storage Service (S3) può essere utilizzato per archiviare oggetti come risorse di gioco, video, immagini, file di registro di testo e altro ancora. S3 è un servizio di storage di oggetti che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni all'avanguardia nel settore.

Se offre più classi di storage che supportano l'accesso frequente e raro ai dati e uno storage di archiviazione conveniente. Per i dati a cui si accede frequentemente durante lo sviluppo, gli studi dovrebbero archiviare gli oggetti in S3 Standard per una bassa latenza e prestazioni di throughput elevate. Per i dati che passano spesso da valori caldi a freddi o viceversa, gli studi dovrebbero prendere in considerazione S3 Intelligent-Tiering. Intelligent-Tiering monitora i modelli di accesso dei dati e li sposta automaticamente al livello di accesso più conveniente.

S3 Express One Zone è ideale per gli studi che necessitano di un throughput elevato, di una bassa latenza e che non hanno problemi a vivere in un'unica zona di disponibilità. Questo consente di replicare i dati su una singola AZ e può migliorare la velocità di accesso ai dati rispetto allo standard S3. Per esigenze di archiviazione approfondita dei dati storici, Amazon offre anche Amazon Glacier. Le classi di storage Amazon Glacier sono create appositamente per l'archiviazione dei dati e offrono prestazioni elevate, flessibilità di recupero e storage di archiviazione a basso costo nel cloud.

Amazon Elastic Block Store può essere utilizzato per archiviare i file binari, i file eseguibili e le configurazioni dei server di gioco necessari per il funzionamento dei server di gioco o degli archivi di risorse. È necessario creare un'istantanea ed eliminare i volumi inutilizzati che non sono collegati a un'istanza. EC2 In questo modo si alleggeriscono i costi di archiviazione sostenuti, riducendo al contempo l'utilizzo di servizi e hardware non necessari.

Passaggi dell’implementazione

- Classificate i dati di gioco per tipo, esigenze di conservazione e frequenza di accesso, etichettando i dati per distinguere tra requisiti di archiviazione a breve e lungo termine.
- Usa Amazon Aurora per i dati transazionali, DynamoDB per i dati dei giocatori in tempo reale, DocumentDB per i dati semistrutturati e per la memorizzazione nella cache a bassa latenza di informazioni di gioco cruciali in termini di tempo. ElastiCache
- Archivia le risorse di gioco, i log e i contenuti generati dagli utenti in Amazon S3, selezionando le classi di storage appropriate (ad esempio, Intelligent-Tiering, One Zone e Glacier) in base ai modelli di accesso e alle esigenze di archiviazione e usa EBS per i binari e le configurazioni dei server di gioco con una gestione regolare delle istantanee.

GAMESUS01-BP02 Utilizza le politiche del ciclo di vita o la scadenza TTL per eliminare giochi, dati utente, file di registro o risorse obsolete non necessari.

Puoi utilizzare tag e tipi di dati per creare policy relative al ciclo di vita o TTL per spostare i dati nell'archivio o rimuoverli completamente dal servizio. Ciò può includere configurazioni temporanee, contenuti archiviati scaduti e registri cronologici che non sono più necessari. La maggior parte dei servizi supporta l'etichettatura.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

Per i dati archiviati in S3, puoi utilizzare le policy del ciclo di vita per spostare i dati su livelli di storage ad accesso e archiviazione non frequenti. In una configurazione del ciclo di vita S3 puoi definire regole per la transizione degli oggetti da una classe di storage a un'altra per risparmiare sui costi di storage. Quando non si conoscono i modelli di accesso degli oggetti o se cambiano nel tempo, è possibile eseguire la transizione degli oggetti alla classe di archiviazione S3 Intelligent-Tiering per ottenere risparmi sui costi in modo automatico.

Amazon S3 supporta un modello a cascata per la transizione tra classi di storage, come mostrato nel diagramma seguente.

È possibile aggiungere operazioni di transizione a una configurazione del ciclo di vita S3 per indicare ad Amazon S3 di eliminare gli oggetti al termine del loro ciclo di vita. Quando un oggetto raggiunge la fine del suo ciclo di vita in base alla configurazione del ciclo di vita, Amazon S3 esegue un'azione di scadenza in base allo stato di S3 Versioning in cui si trova il bucket:

- Bucket senza versione: Amazon S3 mette in coda l'oggetto per la rimozione e lo rimuove in modo asincrono, rimuovendo definitivamente l'oggetto.
- Bucket abilitato al controllo delle versioni: se la versione corrente dell'oggetto non è un marker di eliminazione, Amazon S3 aggiunge un marker di eliminazione con un ID di versione univoco. La versione corrente diventa quindi non corrente e il contrassegno di eliminazione diventa la versione corrente.
- Bucket con versione sospesa: Amazon S3 crea un marker di eliminazione con null come ID di versione. Questo marker di eliminazione sostituisce la versione di un oggetto con un ID di versione nullo nella gerarchia delle versioni, eliminando di fatto l'oggetto.

- Quando si aggiunge una configurazione del ciclo di vita a un bucket, le regole di configurazione si applicano sia agli oggetti esistenti sia a quelli che vengono aggiunti in un secondo momento. Ad esempio, se oggi aggiungi una regola di configurazione del ciclo di vita con un'azione di scadenza che fa scadere gli oggetti con un prefisso specifico 30 giorni dopo la creazione, Amazon S3 metterà in coda per rimuovere gli oggetti esistenti che hanno più di 30 giorni e che hanno il prefisso specificato.

Il Time to Live (TTL) per DynamoDB è un metodo conveniente per eliminare elementi che non sono più pertinenti. Il TTL consente di definire un timestamp per elemento in modo da indicare quando un elemento non è più necessario. DynamoDB elimina automaticamente gli elementi scaduti entro pochi giorni dalla data di scadenza, senza utilizzare il throughput di scrittura.

- Per utilizzare il TTL, è necessario prima abilitarlo su una tabella e poi definire un attributo specifico per archiviare il timestamp di scadenza del TTL. Il timestamp deve essere archiviato nel formato ora epoch Unix con la granularità dei secondi. Ogni volta che un elemento viene creato o aggiornato, è possibile calcolare l'ora di scadenza e salvarla nell'attributo TTL.
- Gli elementi con attributi TTL validi e scaduti possono essere eliminati dal sistema, in genere entro pochi giorni dalla scadenza. È comunque possibile aggiornare gli elementi scaduti in attesa di eliminazione, come anche modificare o rimuovere i relativi attributi TTL. Durante l'aggiornamento di un elemento scaduto, si consiglia di utilizzare un'espressione condizionale per assicurarsi che l'elemento non sia stato successivamente eliminato. Utilizza le espressioni di filtro per rimuovere gli elementi scaduti dai risultati di Scan e Query.
- Gli elementi eliminati funzionano in modo simile a quelli eliminati tramite le tipiche operazioni di eliminazione. Una volta eliminati, gli elementi entrano in DynamoDB Streams come eliminazioni dal servizio anziché dagli utenti e vengono rimossi dagli indici secondari locali e dagli indici secondari globali proprio come le altre operazioni di eliminazione.

Con ElastiCache for Redis puoi controllare l'aggiornamento dei dati memorizzati nella cache utilizzando o scadendo le chiavi memorizzate nella cache. TTLs Trascorso il tempo impostato, la chiave viene eliminata dalla cache e, oltre all'accesso ai dati aggiornati, è necessario accedere all'archivio dati di origine.

- Due principi determinano i modelli appropriati TTLs da applicare e i tipi di modelli di memorizzazione nella cache da implementare. Innanzitutto, è importante comprendere la velocità di variazione dei dati sottostanti. In secondo luogo, è importante valutare il rischio che all'applicazione vengano restituiti dati obsoleti anziché alla controparte aggiornata.

- Con dati dinamici che cambiano spesso, potresti voler applicare dati con una scadenza inferiore TTLs a quella del database principale. In questo modo si riduce il rischio di restituire dati obsoleti e allo stesso tempo di fornire un buffer per scaricare le richieste del database.
- È inoltre importante riconoscere che, anche se si memorizzano nella cache solo i dati per minuti o secondi anziché per periodi più lunghi, applicarli in modo appropriato alle chiavi memorizzate nella cache può portare TTLs a un aumento delle prestazioni e a un'esperienza di gioco complessivamente migliore.

Passaggi dell'implementazione

- Utilizza le policy del ciclo di vita di Amazon S3 per trasferire gli oggetti a livelli di accesso o archiviazione poco frequenti e configura le azioni di scadenza per eliminare gli oggetti non necessari in base alle regole del ciclo di vita.
- Abilita Time to Live (TTL) nelle tabelle DynamoDB per eliminare automaticamente gli elementi scaduti senza consumare il throughput di scrittura, definendo il timestamp di scadenza in Unix epoch time.
- Imposta ElastiCache le chiavi in modo appropriato in base ai tassi di modifica dei dati e alla tolleranza al rischio TTLs per i dati obsoleti, facilitando l'aggiornamento dei dati memorizzati nella cache e una migliore esperienza del giocatore.

Hardware e servizi

GAMESUS02: Come gestisci l'uso delle risorse di calcolo nel backend dei tuoi giochi?

Gli studi dovrebbero sviluppare una strategia di elaborazione che utilizzi un mix di diversi tipi di elaborazione, servizi gestiti e un piano di risparmio per ottimizzare l'utilizzo. È inoltre necessario ottimizzare il modo in cui i server di gioco e i servizi di backend vengono impacchettati per elaborare le istanze per ridurre il numero di risorse non necessarie.

Best practice

- GAMESUS02-BP01 Seleziona servizi gestiti per carichi di lavoro di elaborazione appropriati
- GAMESUS02-BP02 Dimensiona correttamente il tuo calcolo e distribuisci le prestazioni della GPU solo dove necessario

GAMESUS02-BP01 Seleziona servizi gestiti per carichi di lavoro di elaborazione appropriati

Progetta i tuoi servizi di backend di gioco per utilizzare servizi gestiti per carichi di lavoro basati su eventi o con traffico altamente variabile. I servizi gestiti spostano la gestione dell'infrastruttura verso AWS e distribuiscono l'impatto ambientale tra più utenti grazie ai piani di controllo multi-tenant.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

AWS servizi come AWS Lambda, AWS Fargate (Containers) e Amazon Gamelift (orchestrazione di server di gioco) possono eseguire codice, contenitori o orchestrare i server di gioco senza dover gestire l'infrastruttura sottostante. Questi servizi si ridimensionano automaticamente in base alla domanda dei giocatori e ti vengono addebitati solo i costi per le risorse che consumi. Poiché l'infrastruttura sottostante è gestita per tuo conto, puoi concentrarti esclusivamente sui requisiti dei tuoi giochi e dei servizi di backend.

Con AWS Lambda puoi eseguire codice senza effettuare il provisioning o gestire server. Lambda esegue il codice su un'infrastruttura di calcolo ad alta disponibilità ed esegue l'amministrazione delle risorse di calcolo, inclusa la manutenzione di server e sistemi operativi, il provisioning della capacità, il ridimensionamento automatico e la registrazione. Con Lambda, devi fornire il codice in uno dei runtime linguistici supportati da Lambda. Lambda è utile per elaborare gli eventi di gioco, l'autenticazione dei giocatori, l'elaborazione degli acquisti in-game e le richieste di matchmaking. Lambda si ridimensiona automaticamente in base al numero di eventi e può gestire picchi di traffico imprevisti.

AWS Fargateè un motore di elaborazione serverless per contenitori che funziona sia con Amazon Elastic Container Service (ECS) che con Amazon Elastic Kubernetes Service (EKS). AWS Fargate semplifica la concentrazione sulla creazione delle applicazioni riducendo la necessità di fornire e gestire i server, consente di specificare e pagare le risorse per applicazione e migliora la sicurezza attraverso l'isolamento delle applicazioni fin dalla progettazione. Fargate è ideale per i servizi di backend che gestiscono i profili dei giocatori, la gestione dello stato e il matchmaking.

Amazon GameLift è un servizio gestito per la distribuzione, il funzionamento e il ridimensionamento di server di gioco dedicati per giochi multiplayer basati su sessioni. Puoi implementare il tuo primo server di gioco nel cloud in pochi minuti, risparmiando fino a migliaia di ore di progettazione nello sviluppo iniziale del software e riducendo i rischi tecnici che spesso inducono gli sviluppatori a eliminare le funzionalità multiplayer dai loro progetti.

Passaggi dell’implementazione

- Utilizzalo AWS Lambda per carichi di lavoro basati sugli eventi, come l'elaborazione di eventi di gioco, l'autenticazione dei giocatori, gli acquisti in-game e le richieste di matchmaking, sfruttando il ridimensionamento automatico e la gestione senza server.
- Implementa AWS Fargate con ECS o EKS per servizi di backend come profili dei giocatori, gestione dello stato e matchmaking, rimuovendo la gestione dei server e migliorando l'isolamento delle applicazioni.
- Usa Amazon GameLift per distribuire e scalare server di gioco dedicati per giochi multiplayer basati su sessioni, riducendo i tempi di sviluppo e la complessità operativa.

GAMESUS02-BP02 Dimensiona correttamente il tuo calcolo e distribuisci le prestazioni della GPU solo dove necessario

Progetta i server di gioco e il backend per utilizzare in modo efficiente le risorse di elaborazione. L'eccessivo provisioning delle risorse di elaborazione può comportare costi inutili e ridurre al minimo la quantità di risorse inattive o sottoutilizzate. Le istanze GPU devono essere utilizzate per supportare iniziative di sviluppo specifiche, come le ricostruzioni HLOD in Unreal o se i server di gioco le richiedono fin dalla progettazione. Ciò riduce notevolmente l'impatto ambientale e i costi dei carichi di lavoro.

Livello di rischio associato se questa best practice non fosse adottata: elevato

Guida all’implementazione

È necessario ottimizzare i server di gioco e i servizi di backend per utilizzare più tipi di EC2 istanze e il minor numero di istanze necessario. Ciò aumenta il numero di istanze disponibili per soddisfare le tue esigenze durante lo sviluppo o per il lancio dei giochi. Inoltre, dovresti abbinare il tipo di istanza al carico di lavoro specifico che stai distribuendo. Le istanze Compute Optimized supportano un'ampia gamma di casi d'uso, tra cui server di gioco e servizi di backend come il matchmaking. Le istanze ottimizzate per la memoria sono progettate per offrire prestazioni rapide per carichi di lavoro che elaborano set di dati di grandi dimensioni in memoria. Utilizza le istanze GPU se necessario per requisiti di prestazioni elevate, ma non per attività di elaborazione generiche. Se possibile, progetta i tuoi servizi o server di gioco in modo che funzionino su ARM con istanze AWS Graviton. Graviton è il tipo di istanza più performante ed efficiente dal punto di vista energetico disponibile su. AWS Offrono inoltre prestazioni e costi migliori rispetto ai tipi di istanze x86.

Utilizzalo AWS Compute Optimizerper identificare le configurazioni di AWS risorse ottimali, come i tipi di istanze di Amazon Elastic Compute Cloud (EC2), le configurazioni dei volumi Amazon Elastic Block Store (EBS), le dimensioni delle attività dei AWS Fargate servizi Amazon Elastic Container Service (ECS), le AWS Lambda licenze software commerciali, le dimensioni della memoria funzionale e le classi di istanze DB di Amazon Relational Database Service (RDS), utilizzando l'apprendimento automatico per analizzare metriche storiche di utilizzo. Compute Optimizer offre un set e un'esperienza APIs di console per ridurre i costi e aumentare le prestazioni del carico di lavoro consigliando le risorse ottimali per i AWS tuoi carichi di lavoro. AWS

Passaggi dell'implementazione

- Abbina le risorse di elaborazione a carichi di lavoro specifici utilizzando istanze Compute Optimized per server di gioco, istanze ottimizzate per la memoria per set di dati di grandi dimensioni e istanze GPU solo per attività come ricostruzioni HLOD o server di gioco dipendenti dalla GPU.
- Ottimizza l'utilizzo dell'elaborazione implementando le istanze AWS Graviton ove possibile per l'efficienza energetica, migliori prestazioni e risparmi sui costi rispetto alle istanze x86.
- Utilizzalo AWS Compute Optimizer per analizzare lo storico di utilizzo e consigliare le configurazioni più efficienti per EC2 i carichi di lavoro AWS ECS AWS Lambda e Amazon RDS per ridurre i costi e migliorare le prestazioni.

Resources

Consulta le seguenti risorse per saperne di più sulle nostre migliori pratiche relative alla sostenibilità.

- ONU: L'industria del gioco mette in luce le minacce al pianeta
- Medium: Sostenibilità ambientale nello sviluppo di giochi: giocare in modo responsabile per un futuro più verde

Servizi chiave AWS

- Amazon Aurora
- Amazon DynamoDB
- Amazon DocumentDB (compatibile con MongoDB)
- Amazon ElastiCache
- Amazon S3

- Classi di storage Amazon S3
- Classe di storage Amazon S3 Intelligent-Tiering
- Classe di storage Amazon S3 Express One Zone
- Amazon Elastic Block Store
- AWS Lambda
- Amazon Elastic Container Service
- Amazon Elastic Kubernetes Service
- Amazon GameLift
- AWS Compute Optimizer

Conclusioni

I giochi sono progettati per offrire esperienze di intrattenimento a un pubblico globale di giocatori e presentano caratteristiche di utilizzo generalmente imprevedibili e variabili. Games Industry Lens descrive i tipi più comuni di scenari che in genere costituiscono un'architettura di gioco e fornisce una serie di domande e best practice da considerare quando si creano e gestiscono giochi nel cloud. Applicando questo framework all'architettura di gioco, puoi creare giochi affidabili, sicuri, efficienti ed economici nel cloud.

Collaboratori

Le seguenti persone hanno contribuito a questo documento:

- Adam Hatfield, Architetto di soluzioni senior, Amazon Web Services
- Brady Webb, responsabile tecnico degli account, Amazon Web Services
- Bruce Ross — Architetto senior delle soluzioni, Well-Architected Lens Leader, Amazon Web Services
- Caleb Cecil, Architetto associato delle soluzioni, Amazon Web Services
- Carlos Perez, Cloud Optimization Success SA, Amazon Web Services
- Chase Herrington, Account Manager tecnico ESL, Amazon Web Services.
- Chris Blackwell, Architetto delle soluzioni senior, Amazon Web Services
- Corey Ouderkirk, responsabile tecnico degli account senior, Amazon Web Services
- Derek Villavicencio, Prototyping SA, Amazon Web Services
- Erik Ynigo Becerril, Architetto di soluzioni senior, Amazon Web Services
- Grzegorz Ochmanski, architetto senior delle soluzioni, Amazon Web Services
- Hadrian Baron, responsabile tecnico degli account senior, Amazon Web Services
- Ian Armbruster, responsabile delle soluzioni per i clienti senior, Amazon Web Services
- Jed Obray, responsabile tecnico senior, Amazon Web Services
- Khurram Khokhar, responsabile tecnico senior, Amazon Web Services
- Kyle Somers, Senior Manager, Architettura delle soluzioni, Amazon Web Services
- Madhuri Srinivasan, redattore tecnico senior, Well-Architected, Amazon Web Services
- Matthew Wygant, Guida TPM senior, Well-Architected, Amazon Web Services
- Nataliya Godunok, Cloud Optimization Success SA, Amazon Web Services
- Nirav Doshi, Architetto principale delle soluzioni, Amazon Web Services
- Olivia Liddell, architetto di soluzioni, Amazon Web Services
- Randy James, responsabile tecnico principale degli account, Amazon Web Services
- Reou Ando, architetto di soluzioni di gioco, Amazon Web Services
- Richard Raseley, Responsabile tecnico degli account senior, Amazon Web Services
- Sam Patzer, Architetto di soluzioni senior, Amazon Web Services
- Scott Selinger, architetto senior delle soluzioni, Amazon Web Services

- Sean Allen, architetto senior delle soluzioni, Amazon Web Services
- Serge Poueme, Architetto di soluzioni senior, Amazon Web Services
- Stewart Matzek, redattore tecnico senior, Well-Architected, Amazon Web Services
- Trenton Potgieter, Architetto delle soluzioni seniorAI/ML/Analytics, Amazon Web Services

Revisioni del documento

Per ricevere una notifica sugli aggiornamenti del presente whitepaper, iscriviti al feed RSS.

Modifica	Descrizione	Data
Nuova versione dell'obiettivo	L'intero obiettivo è stato aggiornato con nuove linee guida sulle migliori pratiche.	9 dicembre 2025
Pubblicazione iniziale	Whitepaper pubblicato per la prima volta.	19 novembre 2021

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.