Service Name

Amazon GameLift Streams

Service Overview

Amazon GameLift Streams is a fully managed cloud-based game streaming service that enables game publishers to deliver high-definition gaming experiences directly to players' browsers without requiring game downloads or installations. The service uses AWS GPU instances and streaming technology to run games on cloud infrastructure and stream them at up to 1080p resolution and 60 fps to any device with a WebRTC-enabled browser. Publishers can deploy games in minutes with minimal modifications, leveraging AWS global infrastructure to provide low-latency gameplay experiences while maintaining full control over player relationships, branding, and business models.

Key Use Cases

- Direct distribution: Enable instant-play game distribution without app stores or lengthy downloads, reaching players directly through browsers
- Game demos and marketing: Create on-demand interactive game demos and trailers that potential customers can experience immediately
- Secure playtesting: Conduct internal and external game testing without distributing game builds, protecting intellectual property during development
- Legacy game revival: Relaunch back catalog games on modern devices without porting or rebuilding for new platforms
- Virtual worlds and metaverse: Build online interactive experiences and virtual worlds accessible through standard web browsers

Features

- **High-definition streaming**: Delivers gameplay at up to 1080p resolution and 60 frames per second with minimal latency
- **Multi-runtime support**: Supports Windows, Linux, and Proton runtime environments for broad game compatibility
- **Auto-scaling capacity**: Provides both always-on and on-demand capacity with automatic scaling based on player demand
- **Multi-location deployment**: Deploy across multiple AWS regions for optimal latency and global reach
- **Stream performance analytics**: Built-in monitoring through CloudWatch with real-time metrics and logging capabilities
- **WebRTC streaming protocol**: Uses open-standard WebRTC for browser compatibility without plugins or downloads

- **Data channel communication**: Enables bidirectional communication between streaming applications and web clients

Value Proposition

Amazon GameLift Streams eliminates infrastructure complexity while providing enterprise-grade streaming performance. Publishers maintain complete control over player relationships, branding, and monetization without managing servers or scaling concerns:

- The service offers instant market entry with existing games requiring minimal modifications, supporting major game engines including Unity, Unreal, and Godot
- Cost-effective scaling allows paying only for allocated capacity with flexible always-on and on-demand options
- AWS global infrastructure ensures low-latency delivery worldwide, while integrated monitoring and security features provide production-ready reliability from day one

Service Integration

Amazon GameLift Streams deeply integrates with core AWS infrastructure services. Amazon S3 stores game application files and logs, while EC2 G-series GPU instances power the streaming compute resources:

- Amazon CloudWatch provides comprehensive monitoring, metrics, and alarming capabilities
- AWS CloudTrail tracks API activity for security and compliance
- Amazon API Gateway and AWS Lambda enable serverless backend integrations for session management
- AWS Identity and Access Management (IAM) controls access to streaming resources
- The service leverages AWS global network backbone for optimized content delivery and can integrate with Amazon Cognito for player authentication workflows

Onboarding and Offboarding

Getting started involves five key steps: First, set up an AWS account with administrative access and download the GameLift Streams Web SDK. Second, upload game files to Amazon S3 using the default storage class:

- Third, create a GameLift Streams application by specifying the S3 URI and executable path
- Fourth, create a stream group defining hardware configuration, locations, and capacity requirements
- Finally, test the stream directly in the AWS console or integrate with the provided web client samples

- The service provides comprehensive documentation, sample applications, and best practice guides to accelerate deployment from development to production environments

Getting Started Guide:
https://docs.aws.amazon.com/gameliftstreams/latest/developerguide/streaming-process.html

Data Removal

Offboarding requires deleting stream groups first, which automatically releases all allocated compute capacity and terminates active sessions. Applications must be unlinked from all stream groups before deletion. Game files stored in customer-owned S3 buckets remain under customer control and must be deleted separately. Application logs and CloudWatch metrics follow standard AWS retention policies. The service provides clear resource dependencies to ensure complete cleanup and cost cessation upon stream group deletion.

Backup/Restore and Disaster Recovery

Amazon GameLift Streams operates as a stateless streaming service without built-in backup capabilities for game save data or persistent storage. Game applications and files are stored in Amazon S3, which provides 99.999999999% durability and cross-region replication options. Session state and progress must be managed by the game application itself, typically through external databases or cloud save services integrated by the developer.

Backup Capabilities

The service does not provide game state backup capabilities as it focuses on streaming compute resources. Application files stored in S3 inherit S3's durability and versioning features. GameLift Streams does not integrate with AWS Backup service since it manages ephemeral streaming sessions rather than persistent data requiring backup protection.

Disaster Recovery

GameLift Streams supports multi-region deployment for geographic redundancy but does not integrate with AWS Elastic Disaster Recovery. Recovery strategies involve recreating stream groups in alternate regions and updating DNS routing. The stateless nature of streaming sessions means disaster recovery focuses on infrastructure recreation rather than data recovery processes.

Recovery Objectives

Multi-region stream group deployment enables RPO of zero for application files stored in S3 and RTO measured in minutes for recreating streaming infrastructure. Always-on capacity in multiple regions can provide near-instantaneous failover for active streaming

sessions. Customers can achieve aggressive RTO targets by maintaining warm standby capacity across regions, though this increases operational costs for improved availability.

Service Constraints

Service Quotas: https://docs.aws.amazon.com/gameliftstreams/latest/developerguide/quotas.html

FAQ: https://aws.amazon.com/gamelift/streams/faqs/

Technical Requirements

Service Documentation: https://docs.aws.amazon.com/gameliftstreams/latest/developerguide/index.html

User Guide: https://docs.aws.amazon.com/gameliftstreams/latest/developerguide/what-is-service.html

API Reference: https://docs.aws.amazon.com/gameliftstreams/latest/apireference/index.html

Pricing Overview

Please see the AWS UK G Cloud 15 Pricing Document accompanying this service in the Digital Marketplace.

Public Pricing: https://aws.amazon.com/gamelift/streams/pricing/

Cost Optimization

GameLift Streams offers sophisticated capacity management with always-on, on-demand, and target-idle capacity configurations. Always-on capacity provides instant availability but incurs continuous costs, while on-demand capacity scales automatically with usage patterns:

- Target-idle capacity maintains a buffer for quick response while minimizing waste
- Multi-tenancy on high-performance stream classes allows multiple applications per GPU resource
- Proton runtime environment can reduce costs compared to Windows licensing
- Regional deployment optimization and proper capacity planning based on usage patterns are essential for cost control

Savings Considerations

Primary cost optimization involves rightsizing stream capacity based on actual usage patterns rather than peak demand. Using on-demand capacity for variable workloads and always-on only for baseline requirements significantly reduces costs:

- Multi-tenancy capabilities on supported stream classes can halve compute costs per stream
- Choosing appropriate stream classes based on game requirements prevents over-provisioning
- Regional selection impacts pricing, with some regions offering lower costs
- Proper application lifecycle management, including timely deletion of unused applications and stream groups, prevents unnecessary storage and compute charges