

Service Name

Amazon GameLift

Service Overview

Amazon GameLift is a fully managed service for deploying, operating, and scaling dedicated game servers for session-based multiplayer games. Built on AWS global computing infrastructure, GameLift helps deliver high-performance, high-reliability, low-cost game servers while dynamically scaling resource usage to meet worldwide player demand. The service provides automated scaling, patching, and fleet management capabilities, allowing game developers to focus on game development instead of server infrastructure management.

Key Use Cases

- **Session-based multiplayer games:** Hosting dedicated game servers for competitive shooters, battle royale, card games, and fighting games with low latency
- **Global game deployment:** Deploying game servers worldwide across AWS Regions and Local Zones to serve players with optimal connection experiences
- **Auto-scaling game hosting:** Managing fluctuating player demand with automatic capacity scaling and cost optimization using Spot instances

Features

- **Managed Hosting:** Fully managed EC2 and container-based game server hosting with automated deployment, scaling, patching, and fleet management
- **GameLift Anywhere:** Hybrid hosting solution allowing integration of on-premises hardware or other cloud providers with GameLift's session management
- **FlexMatch Matchmaking:** Customizable matchmaking service that automatically connects players based on latency, skill level, and game session attributes
- **Auto Scaling:** Target-based auto scaling that adjusts fleet capacity based on player demand while maintaining cost efficiency
- **Multi-Location Deployment:** Deploy game servers across multiple AWS Regions and Local Zones for global coverage and optimal player experiences
- **Game Session Management:** Comprehensive session lifecycle management including player validation, reconnection support, and session protection

Value Proposition

Amazon GameLift offers significant advantages over self-managed or alternative hosting solutions. It provides up to 70% cost savings compared to on-premises deployments through Spot instances and Graviton processors, while eliminating the complexity of infrastructure management:

- The service delivers enterprise-grade security with DDoS protection, global scale with deployment across AWS Regions and Local Zones, and built-in tools for monitoring, debugging, and matchmaking
- GameLift's deep AWS integration enables seamless connectivity with other services like CloudWatch, SNS, and DynamoDB, while supporting popular game engines including Unity and Unreal Engine through dedicated plugins

Service Integration

Amazon GameLift integrates deeply with core AWS services to provide comprehensive game hosting solutions. It leverages Amazon EC2 for compute infrastructure and Amazon ECS for container deployment, while using Amazon CloudWatch for monitoring, logging, and alarms:

- The service integrates with Amazon SNS and Amazon EventBridge for event notifications, Amazon DynamoDB for player data storage, and Amazon S3 for build artifacts and logs
- Security integrations include AWS Shield for DDoS protection, AWS IAM for access control, and Amazon Cognito for player authentication
- Additional integrations support AWS Lambda for backend processing, Amazon API Gateway for client APIs, and AWS Systems Manager for remote server access

Onboarding and Offboarding

Customers can get started with Amazon GameLift through multiple pathways designed for different expertise levels. The quickest approach uses the game server wrapper for proof-of-concept development, while Unity and Unreal Engine plugins provide guided integration workflows:

- The managed containers starter kit offers automated build and deployment solutions
- Customers must first create an AWS account, set up IAM permissions, and choose deployment regions
- The recommended learning path involves starting small with local development using GameLift Anywhere, understanding core concepts through documentation, building and testing with the preferred integration method, then scaling and optimizing for production deployment

Getting Started Guide:

<https://docs.aws.amazon.com/gameliftservers/latest/developerguide/getting-started-intro.html>

Data Removal

GameLift data removal involves terminating fleets, deleting builds and scripts, and removing game session logs. Customers can terminate active game sessions manually

through the console or programmatically via APIs. Fleet termination stops all instances and removes associated compute resources. Build artifacts stored in GameLift's catalog are deleted when builds are removed, while custom data stored in integrated AWS services like DynamoDB or S3 must be managed separately according to each service's deletion procedures.

Backup/Restore and Disaster Recovery

Amazon GameLift provides disaster recovery capabilities through multi-region deployment and automated failover mechanisms. Game session placement queues can be configured with multiple fleet destinations across different regions for automatic failover. The service maintains game session logs in CloudWatch and S3 for recovery purposes. However, GameLift does not provide direct backup services for game world state data, requiring customers to implement custom backup strategies using integrated AWS services.

Backup Capabilities

GameLift itself does not integrate with AWS Backup service. The service maintains game session logs and fleet configurations, but persistent game data backup must be implemented using external AWS services like S3 for file storage or DynamoDB for structured data. Customers are responsible for implementing backup strategies for game world state and player progress data through integrated services.

Disaster Recovery

GameLift does not directly integrate with AWS Elastic Disaster Recovery service. However, it provides disaster recovery capabilities through its multi-location fleet deployment and game session placement queues that can automatically failover to secondary regions. The service's built-in redundancy across AWS Regions and availability zones provides infrastructure-level disaster recovery without requiring additional DR services.

Recovery Objectives

GameLift helps meet RPO and RTO objectives through multi-region deployment, automated failover via placement queues, and real-time session management. Game session placement can failover to secondary locations within seconds, supporting low RTO requirements. For RPO objectives, customers must implement data persistence strategies using integrated AWS services, as GameLift focuses on session management rather than long-term data storage.

Service Constraints

Service Quotas: <https://docs.aws.amazon.com/general/latest/gr/gamelift.html>

FAQ: <https://aws.amazon.com/gamelift/faq/>

Technical Requirements

Service Documentation:

<https://docs.aws.amazon.com/gameliftservers/latest/developerguide/gamelift-intro.html>

User Guide:

<https://docs.aws.amazon.com/gameliftservers/latest/developerguide/index.html>

API Reference:

<https://docs.aws.amazon.com/gameliftservers/latest/apireference/Welcome.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/pricing/>

Cost Optimization

GameLift offers multiple unique cost optimization strategies specifically designed for game hosting workloads. Spot instances provide up to 85% cost savings compared to On-Demand pricing, while AWS Graviton instances offer improved price-performance ratios:

- The service supports auto-scaling with configurable capacity buffers to prevent over-provisioning, and fleet replacement strategies for maintaining cost-effective AMI versions
- Container packing optimizes resource utilization by running multiple game server processes per instance
- Multi-region placement queues enable cost-optimized session placement based on both latency and hosting costs, while queue priorities can be configured to prefer lower-cost locations

Savings Considerations

Primary cost savings opportunities in GameLift include leveraging Spot instances for up to 85% savings on compute costs, implementing proper auto-scaling policies to avoid over-provisioning during low-demand periods, and using ARM-based Graviton instances for better price-performance. Container deployment enables higher resource utilization through better packing efficiency:

- Geographic cost optimization through multi-region strategies allows routing to lower-cost locations when latency permits
- The service's pay-per-use model eliminates upfront infrastructure investments, while managed services reduce operational overhead costs compared to self-managed hosting solutions