

Service Name

Amazon GameLift Servers

Service Overview

Amazon GameLift Servers is a fully managed service for deploying, operating, and scaling dedicated game servers for session-based multiplayer games in the cloud. Built on AWS global computing infrastructure, it delivers high-performance, high-reliability game servers while dynamically scaling resource usage to meet worldwide player demand. The service handles server infrastructure provisioning, scaling, session management, and provides comprehensive monitoring capabilities. Amazon GameLift Servers offers flexible hosting options including managed EC2 hosting, managed container hosting, self-managed Anywhere hosting, and hybrid solutions to accommodate different deployment models and existing infrastructure.

Key Use Cases

- **Session-based multiplayer games:** First-person shooters, MOBAs, fighting, racing, and sports games that have defined game sessions with start and end times
- **Global game server deployment:** Deploying game servers across multiple AWS regions and Local Zones worldwide to minimize player latency and improve game experience
- **Cost optimization for game hosting:** Using Spot instances, auto-scaling, and hybrid topologies to reduce infrastructure costs while maintaining performance

Features

- **Global Deployment:** Deploy hosting in AWS Regions and Local Zones worldwide with multi-location placement queues for optimal player experiences
- **Automatic Scaling:** Target-based auto scaling with capacity buffers to handle player demand fluctuations and sudden influxes
- **FlexMatch Matchmaking:** Advanced matchmaking service supporting 2-200 concurrent players with custom rules, team-based matches, and backfilling
- **Game Session Management:** Automated game server process management, session placement, and player session validation with protection features
- **Comprehensive Monitoring:** Real-time metrics, CloudWatch integration, server logs, event tracking, and performance analysis tools
- **Flexible Hosting Options:** Managed EC2, managed containers, Anywhere hosting, and hybrid solutions with Linux/Windows support
- **Cost Optimization:** Spot instances with up to 85% savings, Graviton instances, and intelligent placement algorithms

- **Security & Reliability:** DDoS protection, TLS encryption, VPC integration, and enterprise-grade security features

Value Proposition

Amazon GameLift Servers enables developers to focus on game development rather than server infrastructure management by providing a fully managed hosting solution that can scale to support 100 million concurrent players. The service offers significant cost savings of up to 70% compared to on-premises deployments through features like Spot instances, auto-scaling, and intelligent resource allocation:

- It delivers enterprise-grade reliability and security with global reach across AWS regions and Local Zones, ensuring low-latency gameplay for players worldwide
- The service integrates seamlessly with popular game engines like Unity and Unreal Engine, and supports various deployment models from fully managed to self-managed hosting, providing flexibility for different organizational needs and existing infrastructure

Service Integration

Amazon GameLift Servers deeply integrates with the AWS ecosystem for comprehensive game hosting solutions. Core integrations include Amazon EC2 for compute infrastructure, Amazon ECS and EKS for containerized deployments, and Amazon CloudWatch for monitoring and alerts:

- The service leverages Amazon DynamoDB and Amazon S3 for game state persistence and data storage, Amazon Cognito for player authentication, and AWS Lambda with Amazon API Gateway for custom game features
- Additional integrations include Amazon Kinesis for game analytics, Amazon Chime SDK for voice chat capabilities, Amazon Elastic Load Balancing for traffic distribution, and AWS CloudFormation for infrastructure as code deployments
- The service also utilizes AWS Identity and Access Management (IAM) for security and access control, and AWS Shield for DDoS protection

Onboarding and Offboarding

Customers can get started with Amazon GameLift Servers through multiple pathways designed for different experience levels and requirements. The quickest approach uses the game server wrapper for rapid proof-of-concept development, while more comprehensive options include Unity and Unreal Engine plugins with guided workflows, and starter kits for managed containers:

- The service offers a free tier with 3,000 game sessions and 500,000 server connection minutes monthly for 12 months
- New users must create an AWS account, set up IAM permissions, and select their preferred region

- The recommended learning path involves starting small with sample games, understanding core concepts, choosing an architecture (managed EC2, containers, Anywhere, or hybrid), building and testing the integration, then scaling and optimizing for production deployment

Getting Started Guide:

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

Data Removal

Amazon GameLift Servers provides straightforward data removal processes during offboarding. Game session logs and performance metrics are automatically retained according to configured retention policies. Build catalog data including game server builds and scripts can be deleted through the console or API. Fleet instances and associated data are removed when fleets are deleted, with no persistent data storage on individual instances. Customer-managed persistent game data stored in integrated AWS services like DynamoDB or S3 must be separately managed and deleted according to those services' data removal procedures.

Backup/Restore and Disaster Recovery

Amazon GameLift Servers inherits AWS infrastructure resilience with built-in redundancy across multiple Availability Zones within regions. The service automatically creates Point-in-Time snapshots of EBS volumes every 10 minutes for the last hour, hourly for 24 hours, and daily for 7 days. Game session data and player progress persistence depends on integration with external AWS services like DynamoDB and S3 which provide their own backup capabilities. Fleet configurations and build artifacts are stored redundantly across AWS infrastructure, ensuring availability during infrastructure failures.

Backup Capabilities

Amazon GameLift Servers does not integrate directly with AWS Backup service as it focuses on session-based gameplay rather than persistent data storage. However, individual fleet instances include 50GB EBS General Purpose SSD volumes with automatic snapshot capabilities:

- The build catalog stores game server builds redundantly across AWS infrastructure
- Persistent game data backup depends on integration with AWS Backup-supported services like DynamoDB, S3, and RDS for game state persistence, player profiles, and analytics data

Disaster Recovery

Amazon GameLift Servers does not directly integrate with AWS Elastic Disaster Recovery as it manages ephemeral game sessions rather than persistent infrastructure. The service

provides inherent disaster recovery through multi-region deployment capabilities and automatic failover mechanisms:

- Game hosting fleets can be deployed across multiple AWS regions and Local Zones, with game session placement queues automatically routing players to available resources during outages
- Fleet replacement and automated deployments enable rapid recovery of hosting capacity

Recovery Objectives

Amazon GameLift Servers helps achieve low RTO objectives through automatic scaling and multi-region deployment capabilities that can rapidly restore game hosting capacity. Game session queues provide automatic failover to alternative fleets and regions within seconds. RPO objectives are primarily relevant for persistent game data stored in integrated services like DynamoDB and S3, which provide their own recovery mechanisms. The service's session-based nature means most game data is ephemeral, focusing recovery on restoring hosting availability rather than data recovery.

Service Constraints

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

FAQ: <https://aws.amazon.com/gamelift/servers/faqs/>

Technical Requirements

Service Documentation:

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

User Guide:

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

API Reference:

<https://docs.aws.amazon.com/gameliftservers/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/servers/pricing/>

Cost Optimization

Amazon GameLift Servers offers several unique cost optimization features designed specifically for game hosting workloads. Spot instances provide up to 85% cost savings

compared to On-Demand pricing, with intelligent placement algorithms that minimize interruptions during active game sessions:

- The service supports AWS Graviton instances for improved price-performance ratios and automatic scaling that can reduce capacity to zero when not in use
- Multi-region deployment strategies optimize costs by selecting the most cost-effective regions while maintaining player experience
- Container packing improves resource utilization by running multiple game server processes efficiently
- The service includes detailed cost monitoring through CloudWatch integration and AWS Cost Explorer, with specific metrics for game hosting optimization like percent idle instances and resource utilization tracking

Savings Considerations

Primary cost savings opportunities include leveraging Spot instances for up to 85% reduction in compute costs, implementing auto-scaling policies to match capacity with player demand, and using Linux instances which typically cost less than Windows alternatives. Game developers can save significantly by selecting appropriate instance types matched to their game requirements rather than over-provisioning resources:

- Multi-location fleets enable cost optimization by choosing regions with lower pricing while maintaining acceptable latency
- The service's pay-per-use model eliminates upfront infrastructure investments and long-term commitments
- Container-based deployments improve resource efficiency through better packing density
- Strategic use of minimum capacity settings prevents unnecessary scaling during low-demand periods, while maximum limits prevent runaway costs during traffic spikes or DDoS attacks