Amazon GameLift Streams Gen6 Stream Classes

Private Beta Release Documentation

Purpose

This document provides instructions for selected Amazon GameLift Streams customers to participate in the private beta for new Gen6 stream classes. The Gen6 stream classes are built on NVIDIA L4 Tensor Core GPUs and 3rd generation AMD EPYC processors (AMD EPYC 7R13), offering new high performance options and new multi-tenancy options for cost-effective streaming of well-optimized or low-fidelity games.

Important: This beta is intended for testing and evaluation purposes only. Beta resources are not suitable for production workloads and will be scaled down after the beta period ends.

Prerequisites

Before you begin, ensure that you have:

- An active Amazon GameLift Streams account with beta access
- AWS CLI installed and configured with appropriate permissions
- An application uploaded to Amazon GameLift Streams

Gen6 Stream Classes Overview

The Gen6 stream classes use NVIDIA L4 Tensor Core GPUs and are available in two underlying instance types:

G6.2XLARGE INSTANCE SPECIFICATIONS

- **CPUs:** 8 CPUs (3rd generation AMD EPYC 7R13 processors)
- **Memory:** 32 GiB
- **GPU:** 1 NVIDIA L4 Tensor Core GPU
- **GPU Memory:** 24 GiB

G6.4XLARGE INSTANCE SPECIFICATIONS

- **CPUs:** 16 CPUs (3rd generation AMD EPYC 7R13 processors)
- **Memory:** 64 GiB
- **GPU:** 1 NVIDIA L4 Tensor Core GPU
- **GPU Memory:** 24 GiB

AVAILABLE STREAM CLASSES

Stream Class	Tenancy Ratio	EC2 Instance Type	Usage Scenarios
gen6n_small	1:12	g6.4xlarge	High multi-tenancy option for low-fidelity games
gen6n_medium	1:4	g6.2xlarge	Balanced multi-tenancy for optimized games
gen6n_pro	1:1	g6.4xlarge	Dedicated resources with more CPUs for the most demanding Proton or Linux games
gen6n_pro_win2022	1:1	g6.4xlarge	Windows Server 2022 with dedicated resources and more CPUs for the most demanding Windows games

PRICING

(per hour of capacity)

Stream Class	us-west-2 (Oregon)	us-east-2 (Ohio)	us-east-1 (N. Virginia)	eu-central-1 (Frankfurt)	ap-northeast-1 (Tokyo)
gen6n_small	$0.1599	$0.1599	$0.1599	$0.1930	$0.1919
gen6n_medium	$0.3848	$0.3848	$0.3848	$0.5062	$0.4961
gen6n_pro	$1.7863	$1.7863	$1.7863	$2.3496	$2.3028
gen6n_pro_win2022	$2.7799	$2.7799	$2.7799	$3.3947	$4.3277

Location Options: Gen6 stream classes are available in all locations supported by Amazon GameLift Streams except eu-west-1 (Europe (Ireland)).

Runtime Options: The `gen6n_pro_win2022` stream class supports the Microsoft Windows Server 2022 runtime. All other stream classes in this beta use Ubuntu 22.04 LTS for games using the Linux or Proton runtimes.

Tenancy Ratio: The tenancy ratio (1:x) indicates how many streams can run simultaneously on each cloud compute resource (EC2 instance). Stream classes with a 1:1 ratio of EC2 instances to number of concurrent streaming applications are referred to as *single-tenant* stream classes, and do not have to share resources with other streaming applications on an instance. Each stream in a single-tenant stream class runs on its own instance. Stream classes with 1:2 ratios or higher are referred to as *multi-tenant* stream classes. The compute resources of multi-tenant stream classes (such as CPU, GPU, and memory) are shared among all streaming applications on a particular instance.

Performance Considerations:
For high-performance games requiring full access to the dedicated GPU hardware, we recommend choosing stream classes with 1:1 ratios.

You can experiment running your game on multi-tenant stream classes to explore more cost-effective pricing options. The higher tenancy ratios might provide adequate streaming performance for:

- Well-optimized games and applications
- Low-fidelity content (e.g., 30 fps, 2D games)
- Applications that can efficiently share compute resources

The gen6n_small and gen6n_pro stream classes can help reduce CPU utilization bottlenecks compared to the gen6n_medium stream class due to their additional CPU cores.

Creating Stream Groups with Gen6 Stream Classes

Use the AWS CLI `create-stream-group` command to create a stream group with Gen6 stream classes. Note that when specifying the stream capacity in stream groups with multi-tenant stream classes, the capacity must be a multiple of the tenancy. For example, the `gen6n_medium` stream class has a multi-tenancy of 4. That means each compute resource that gets allocated in your stream group can stream to 4 clients. Therefore, the capacity you request must be in multiples of 4.

EXAMPLE 1: CREATING A STREAM GROUP WITH GEN6N_PRO

With a single-tenant stream class, you can specify any whole number capacity (up to your per-location limit for this beta).

```
aws gameliftstreams create-stream-group \
  --description "Test gen6n_pro" \
  --default-application-identifier a-9ZY8X7Wv6 \
  --stream-class gen6n_pro \
  --location-configurations '[{"LocationName": "us-east-2", "AlwaysOnCapacity": 3}]'
```

EXAMPLE 2: CREATING A STREAM GROUP WITH GEN6N_SMALL

With a multi-tenant stream class, you must specify capacities in multiples of the tenancy. For example, the requested capacity for `gen6n_small` must be in multiples of 12.

```
aws gameliftstreams create-stream-group \
  --description "Test gen6n_small" \
  --default-application-identifier a-9ZY8X7Wv6 \
  --stream-class gen6n_small \
  --location-configurations '[{"LocationName": "us-east-2", "AlwaysOnCapacity": 24}]'
```

Beta Stream Class Quotas

During this private beta, each AWS account is given an initial quota of 5 Gen6 GPUs (instances) per location across all Gen6 stream classes. We recommend starting with lower capacity values and scaling up as needed within your limit.

As a reminder of how stream class service quotas and stream capacity work together, in this beta the quota specifies the total number of Gen6 GPUs that you can request *per location* across *all stream groups* in your account. Each GPU can host a number of streams equal to their tenancy.

EXAMPLE OF QUOTAS AND CAPACITY USAGE

Assume that you're starting with a quota of 5 Gen6 GPUs per location. If you were to create the stream group in Example 1, using the `gen6n_pro` stream class with 3 capacity in `us-east-2`, you would have 2 remaining Gen6 GPUs in `us-east-2` because this stream class has 1:1 tenancy — at 1 stream per GPU you need 3 GPUs for 3 capacity.

Next, if you create the stream group in Example 2, using the `gen6n_small` stream class with 24 capacity in `us-east-2`, that would leave you with 0 Gen6 GPUs remaining in `us-east-2` because this stream class has 1:12 tenancy — at 12 streams per GPU you need 2 GPUs for 24 capacity.

At this point, while you don’t have any remaining Gen6 GPUs in `us-east-2`, you still have 5 Gen6 GPUs available in other locations that you could add to either of the stream groups you just created, or to a new stream group.

Beta Support and Guidelines

SUPPORT DURING BETA

Our goal during the private beta is to gather feedback on the improved price-performance benefits of Gen6 stream classes and validate the service's features and APIs. While we will schedule regular check-ins throughout the beta period, please don't hesitate to reach out with any questions, comments, blocking issues, or other concerns.

BETA DURATION AND RESOURCE MANAGEMENT

Keep the following points in mind regarding beta resources:

- **Not for production:** Beta resources must not be used for production workloads.
- **Resource retention:** Beta resources can be scaled down after the beta period ends.
- **Migration path:** There will be no migration path to generally available Gen6 stream classes post beta. You will need to create new stream groups to use the Gen6 stream classes when they become generally available. Stream class names might change after beta.