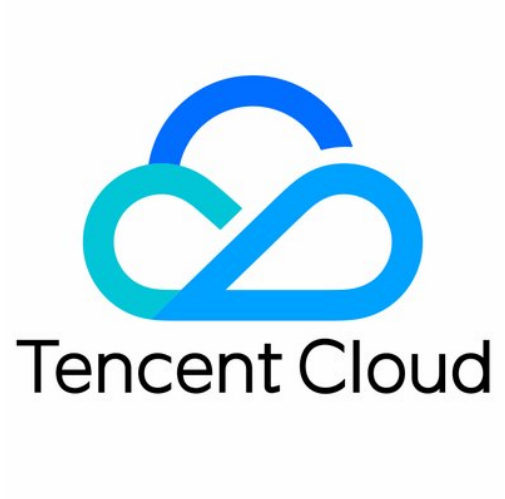
# Game Server Elastic-scaling

# Getting Started

# Product Documentation



# Contents

# Getting Started
# Demo

Last updated : 2022-05-06 10:24:04

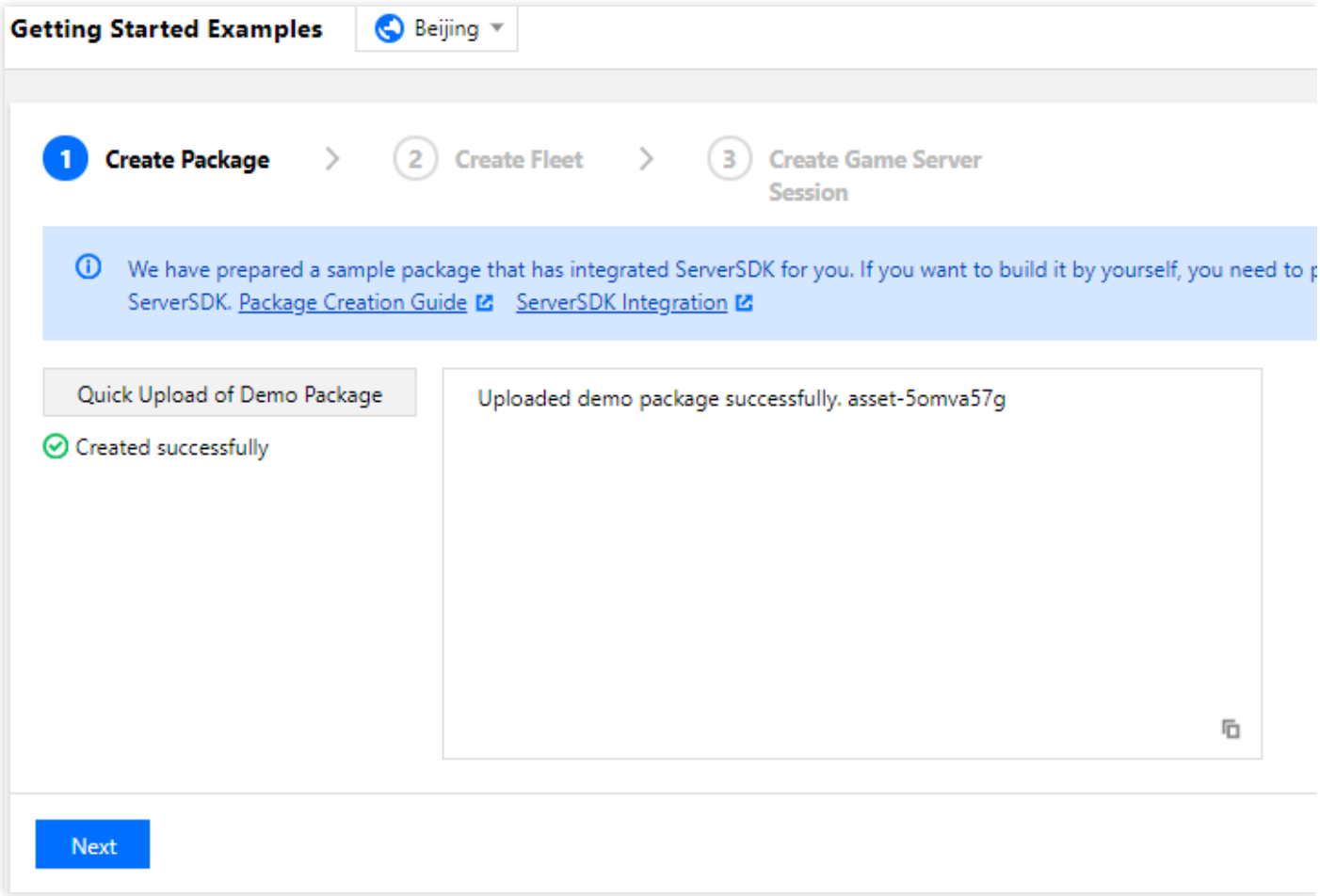
## Overview

This document describes how to use the "Demo" to get started with GSE, a game server hosting service.

## Directions

### Step 1: uploading demo package

1. Log in to the GSE Console and click **Demo** in the left sidebar.
2. Select the service region in the top-left corner and click **Quick Upload of Demo Package**. After the message indicating that the package has been uploaded successfully is displayed, click **Next**.




Note :

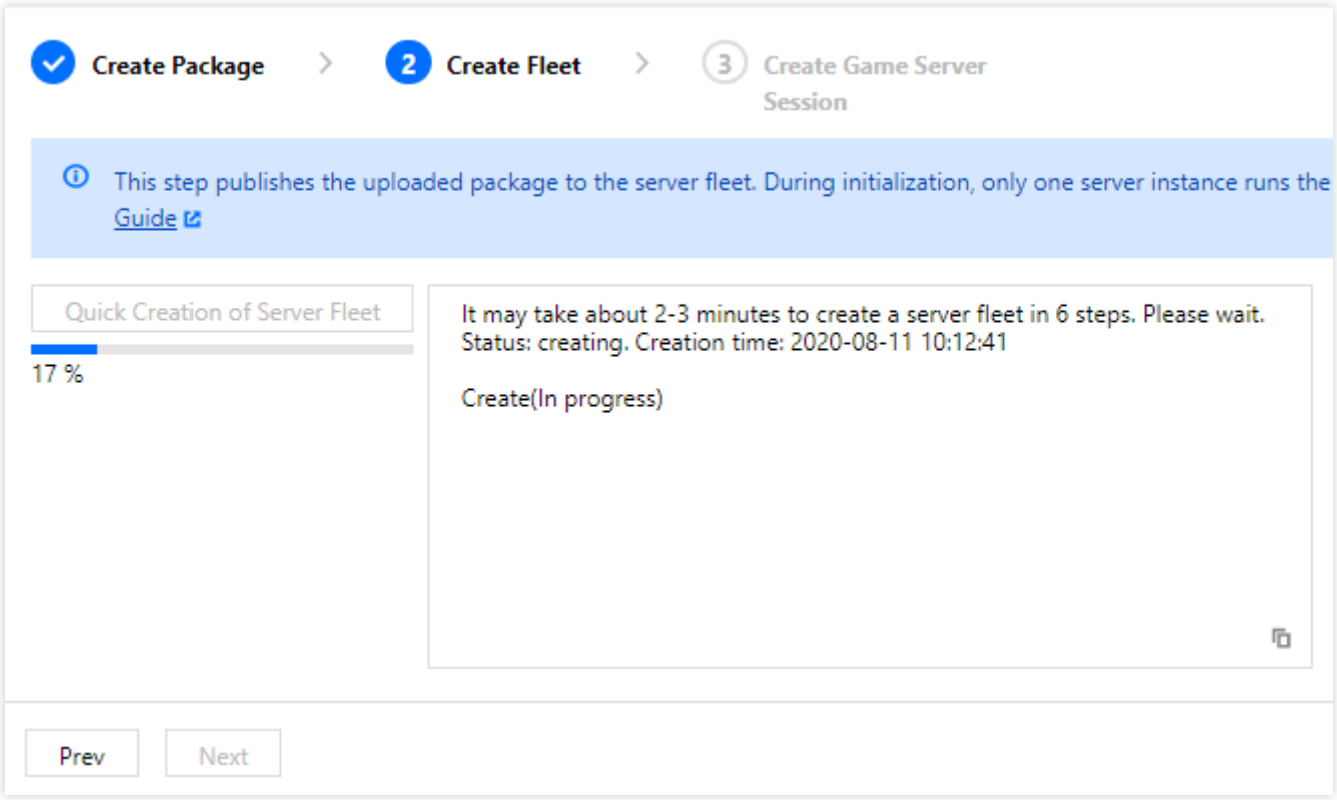
- The demo package provided by GSE has already integrated the gRPC framework through which the game process communicates with GSE.
- If you want to create on your own, please see Creating Code Packages.

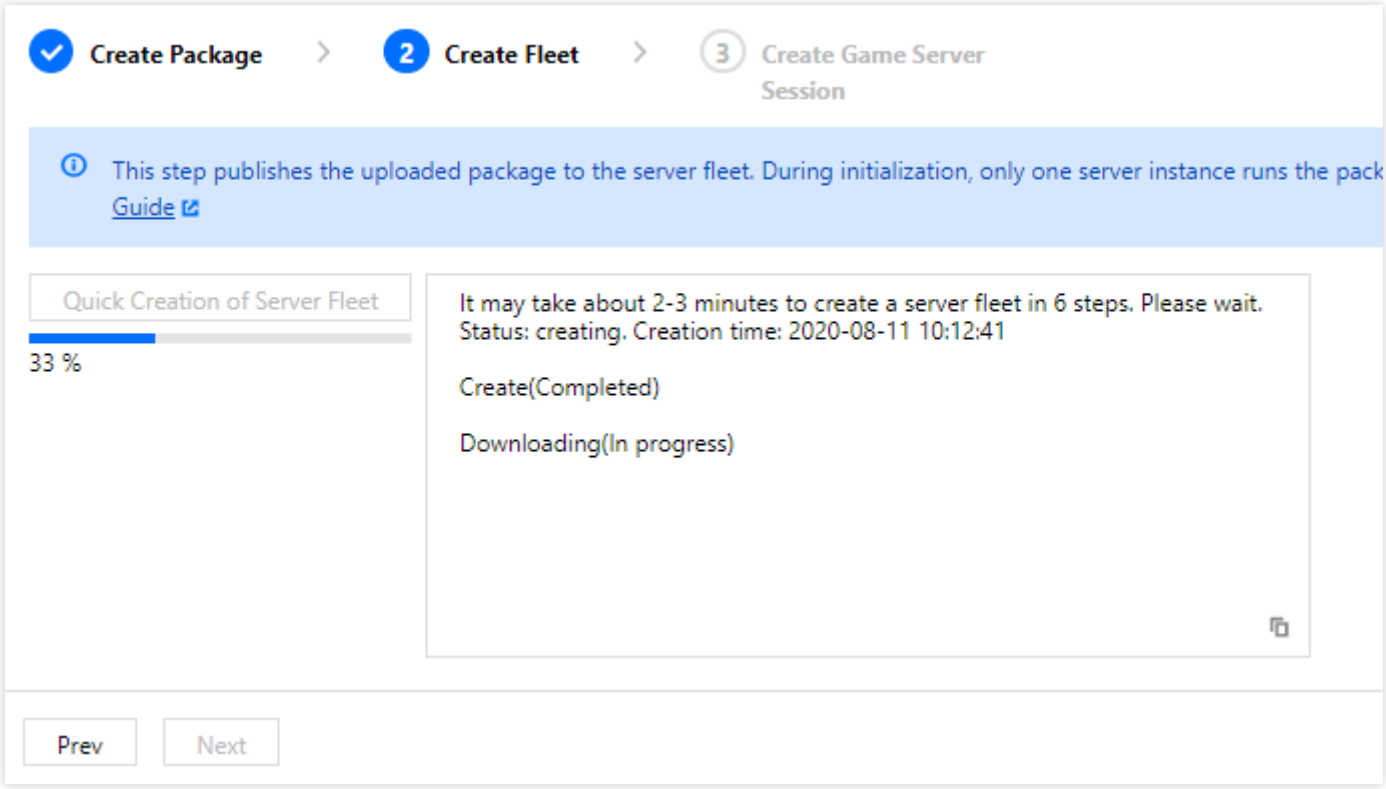
### Step 2: creating a server fleet

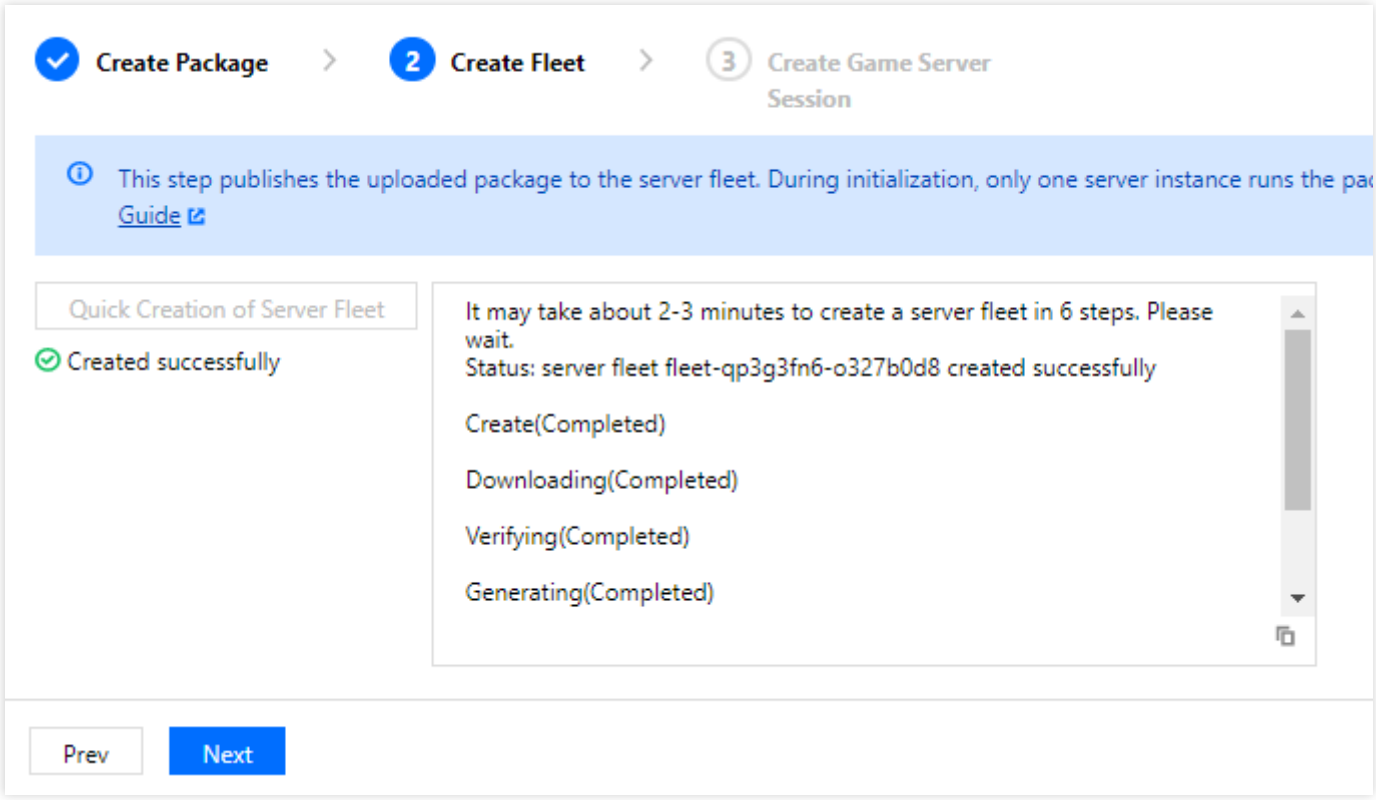
Click **Quick Creation of Server Fleet**, and you will see how the creation status changes, such as “Create (In progress)” and “Downloading (In progress)”. When “Created successfully” is displayed, click **Next**.

- To create a server fleet needs a “Completed” status for all 6 steps, namely, Create (Completed), Downloading (Completed), Verifying (Completed), Generating (Completed), Activating (Completed), and Active (Completed).

- **Status**: creating, or server fleet XXX created successfully.





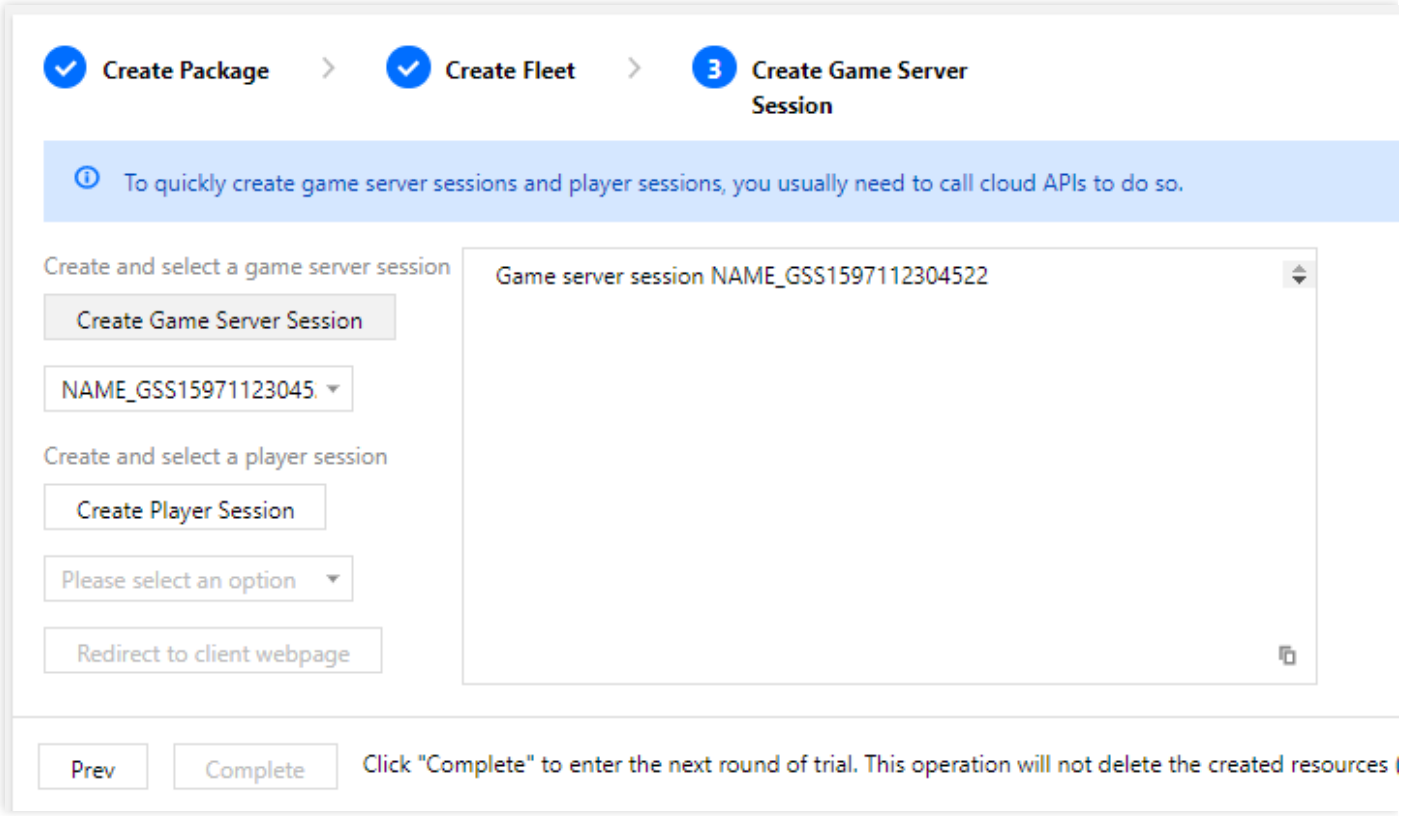




Note :

- Your demo package is deployed onto the server fleet as it is being created.
- A server fleet consists of a group of servers capable of auto-scaling, so the demo package can be deployed globally with ease.
- If you want to create on your own, please see Creating Server Fleets.

### Step 3: creating a game server session and a player session

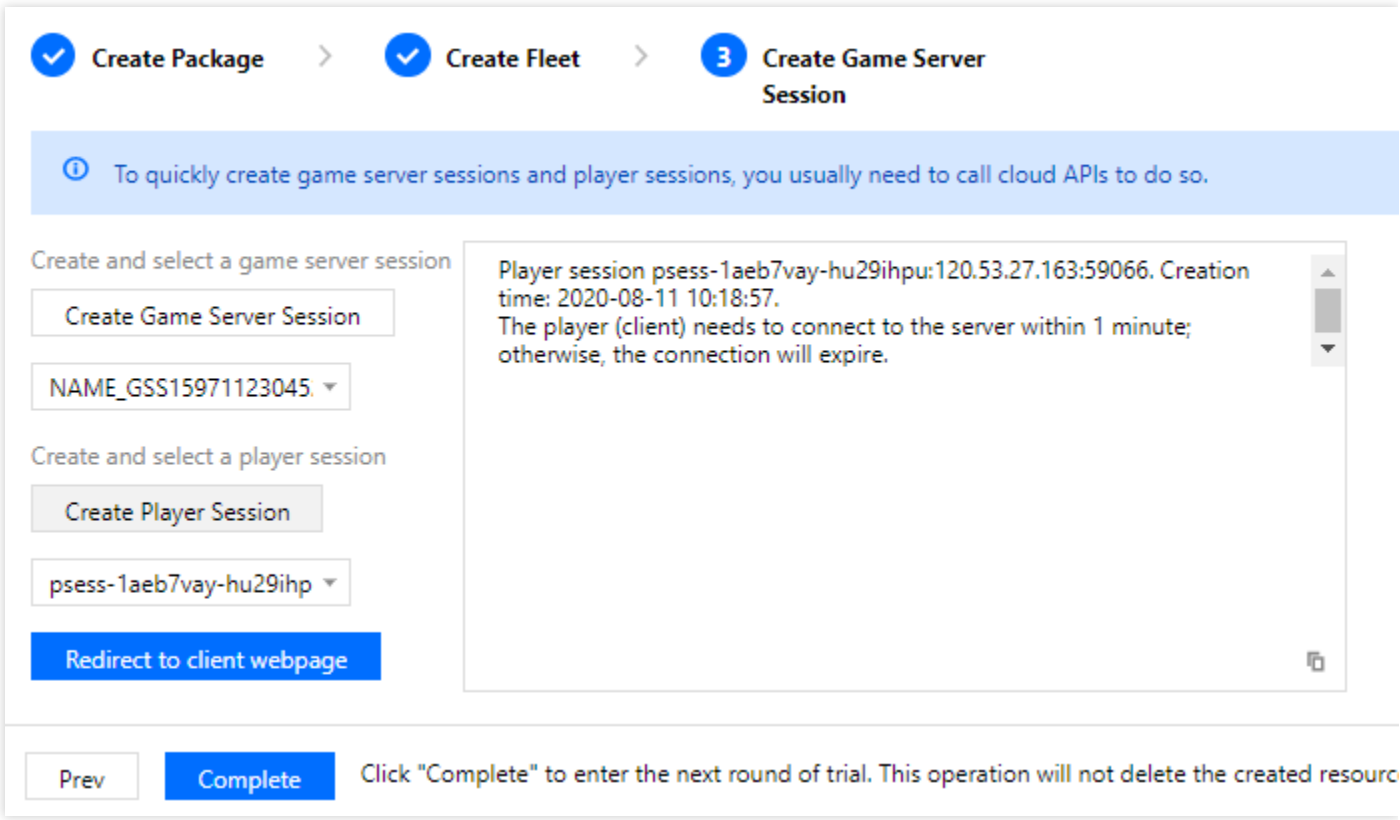
- Click **Create Game Server Session**, and a message will be displayed indicating that a game server session has been created successfully.




Note :

- This operation calls the `CreateGameServerSession` TencentCloud API so that GSE will create a game server session and assign it a service process.
- If you want to create on your own, please see the API document CreateGameServerSession.

- Click **Create Player Session**, and a message indicating that a player session has been created successfully will be displayed.




Note :

- This operation calls the `JoinGameServerSession` TencentCloud API, so that GSE will create a player session and add the player to a game server session.
- If you want to create on your own, please see the API document JoinGameServerSession.

### Step 4: connecting client to the game server

Click **Redirect to client webpage** to access the page for connecting the client to the game server. Click **Connect**, and a message indicating that the server has been connected successfully will be displayed.

Note :

- After creating the player session, the player (client) needs to connect to the server within 1 minute; otherwise, the connection will expire.
- This demo package is a chat service. When multiple players connect to the server, they can chat with each other.

The above four steps simulate the entire GSE integration process. For more information, please see Development Guide.

# Auto Scaling

Last updated : 2020-08-04 11:30:42

## Overview

This document describes how to implement auto scaling through a server fleet.

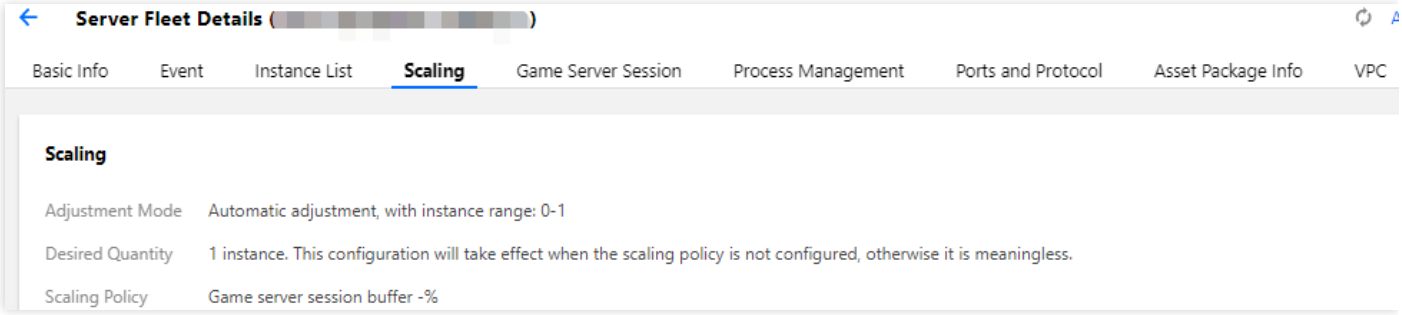
## Prerequisites

You have completed the steps in the Demo.

## Directions

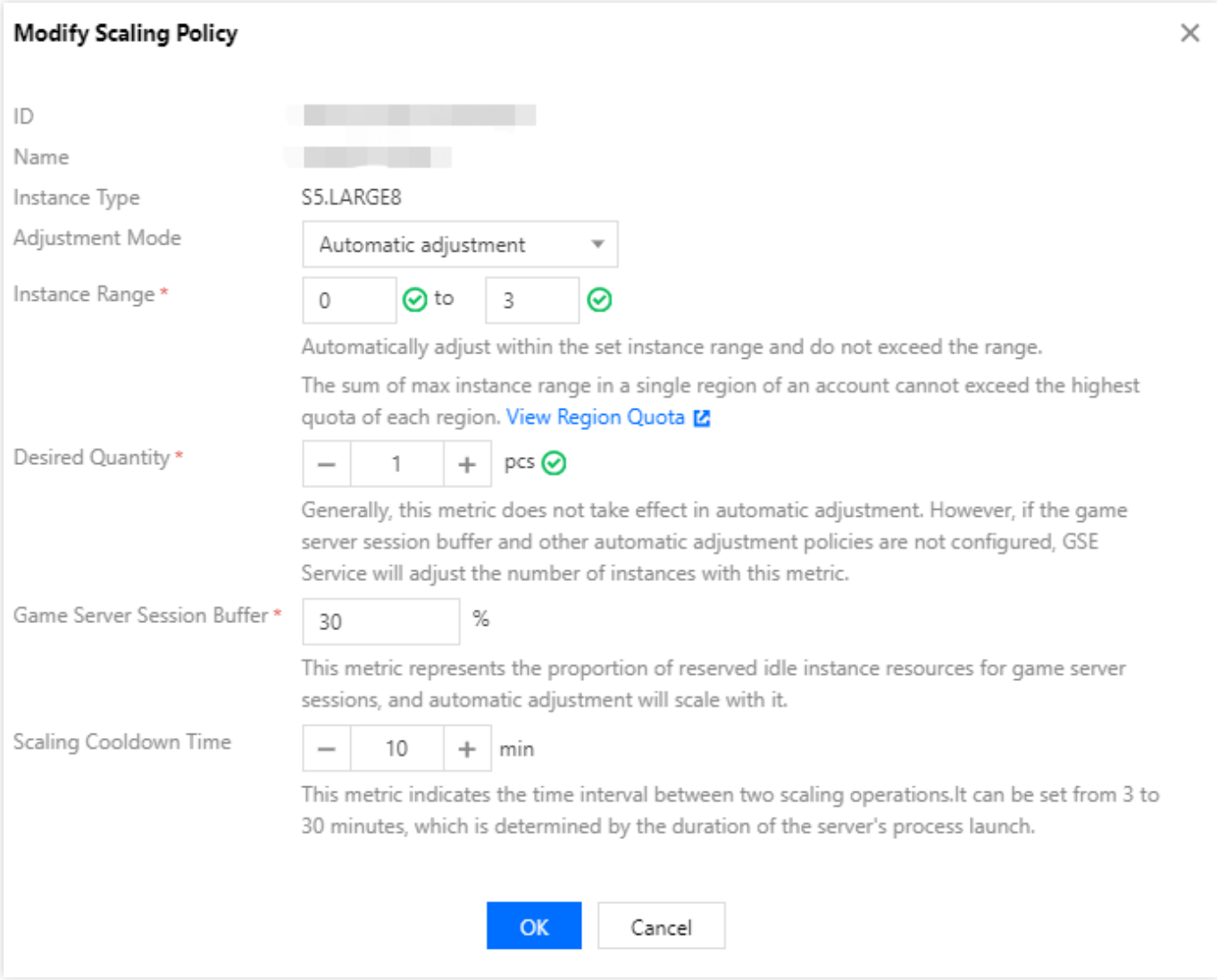
### Modifying the scaling configuration and the number of processes

1. Log in to the GSE Console and click **Fleet** on the left sidebar.
2. Click the ID of the server fleet created in the "Demo" to enter the fleet details page. Click the **Scaling** tab to view scaling details.




3. Click **Modify** in the top-right corner to modify the scaling configuration as follows:
   i. Select "Automatic adjustment" as the adjustment mode.
   ii. Set "Instance Range" to 0–3 so that there will be room for expansion.
   iii. Set the game server session buffer to 30%, i.e., when the number of game battles (sessions) loaded on the server exceeds 70% of the limit, expansion will be performed.

iv. After making the modifications, click **OK**.

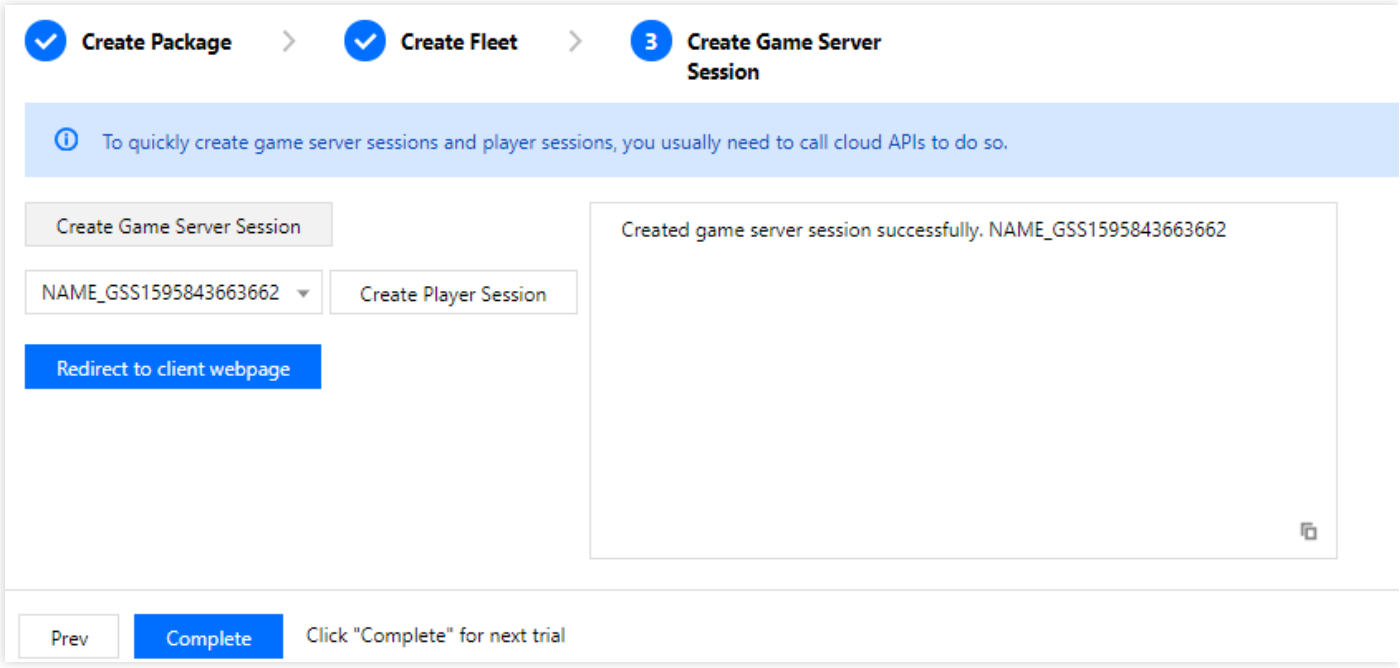



**Note :**

- Game server session buffer = number of available game server sessions / maximum number of game server sessions
  = (maximum number of game server sessions - number of active game server sessions) / maximum number of game server sessions.
- If the game server session buffer is configured as 30%, expansion will be performed when the available game server sessions are below 30%; otherwise, reduction will be performed.

## Creating game server session and observing expansion result

1. In the console, click **Demo** on the left sidebar, complete the first three steps in the Demo, and click **Create Game Server Session** for seven times to create eight game server sessions and trigger expansion.

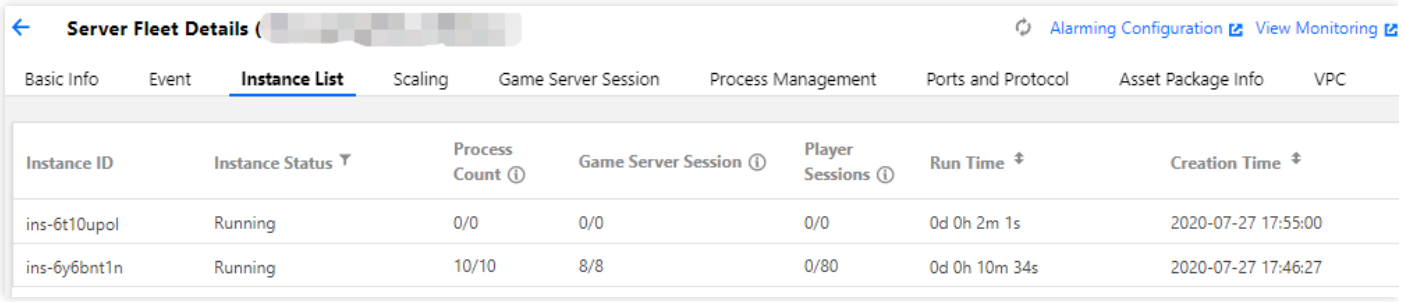



**Note :**

- In GSE Console > Demo, one server can sustain up to 10 game server sessions by default. Therefore, when the server sustains seven game server sessions, the game server session buffer will be number of available game server sessions / maximum number of game server sessions = (maximum number of game server sessions - number of active game server sessions) / maximum number of game server sessions = (10 - 7) / 10 = 30%.
- Therefore, you need to create eight game server sessions at least to trigger expansion.

2. Click **Fleet** on the left sidebar, and select the ID of the created server fleet to enter the fleet details page. Now, click the **Instance List** tab and observe. After two minutes, you will see that

the number of instances is increased to two.

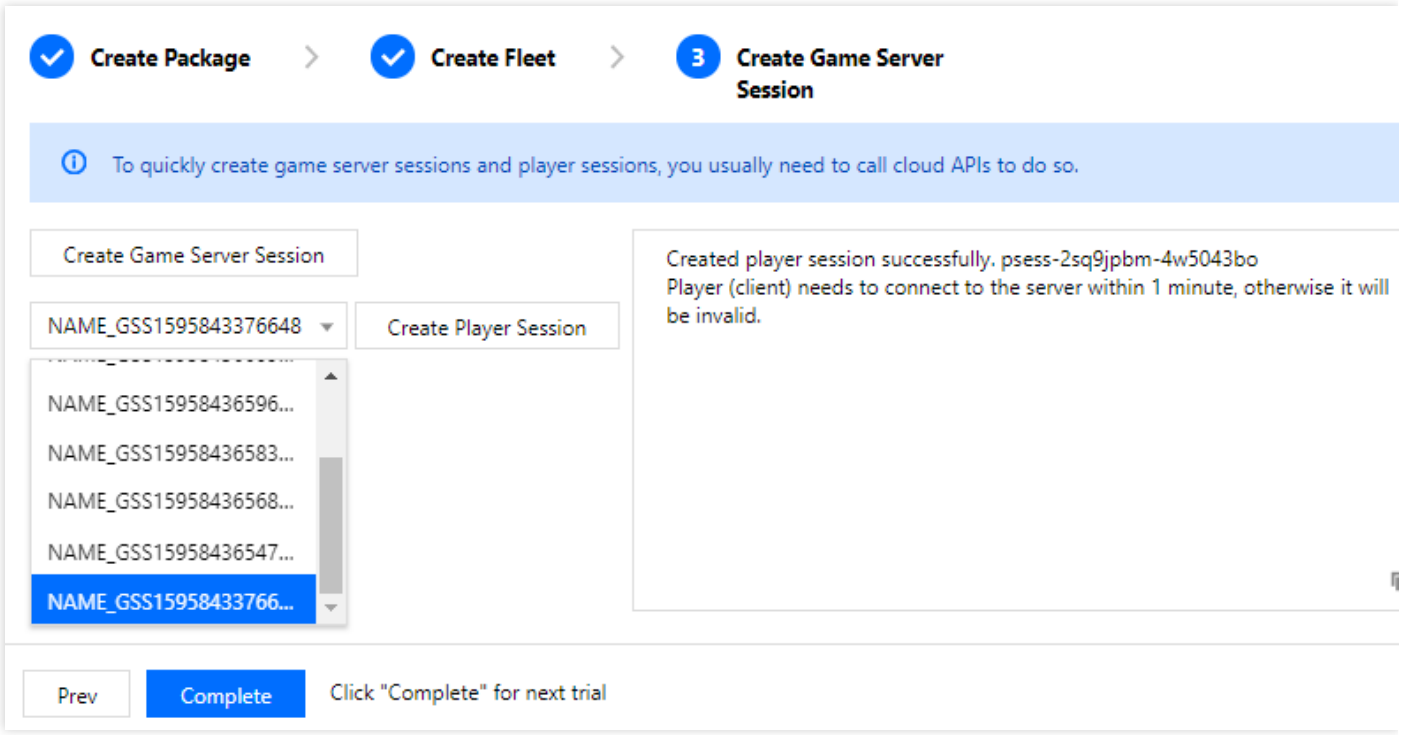



**Note :**

After the creation, **do not** click **Complete** for next round of trial. Instead, you will still need the above configuration for subsequent reduction steps.

## Ending game server session and observing reduction result

1. In the console, click **Demo** on the left sidebar and proceed with the subsequent steps after the above expansion. After selecting each game server session, click **Create Player Session** once to create a player session.

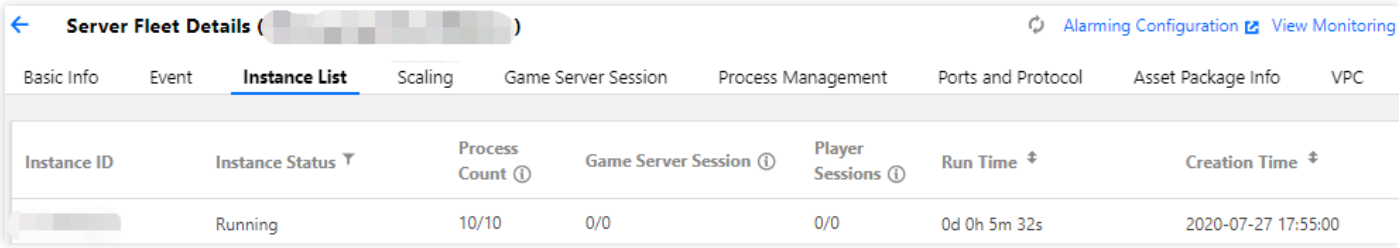



2. Click **Redirect to client webpage** to enter the client page. Click **Connect** to successfully connect to the server. Click **End Game Session** to end the game server session. Repeat steps 1 and 2 once to end 2 game server sessions and trigger reduction.

**Note :**

- Game server session buffer = number of available game server sessions / maximum number of game server sessions = (maximum number of game server sessions - number of active game server sessions) / maximum number of game server sessions = (20 - 6) / 20 = 70%.
- As there are only six active game server sessions left now, making the buffer increased to 70%, reduction is triggered (above 30%).
- Currently, if you close your client webpage, the previously created player sessions will not be able to reconnect to your client. In this case, you have to create a new player session for reconnection in order to close the game server session.

3. Click **Fleet** on the left sidebar, select the ID of the created server fleet to access the fleet details page, and select **Instance List**. Observe the instance quantity. After two minutes, you will see the number of servers decreases to two.



# Zero Downtime Updates

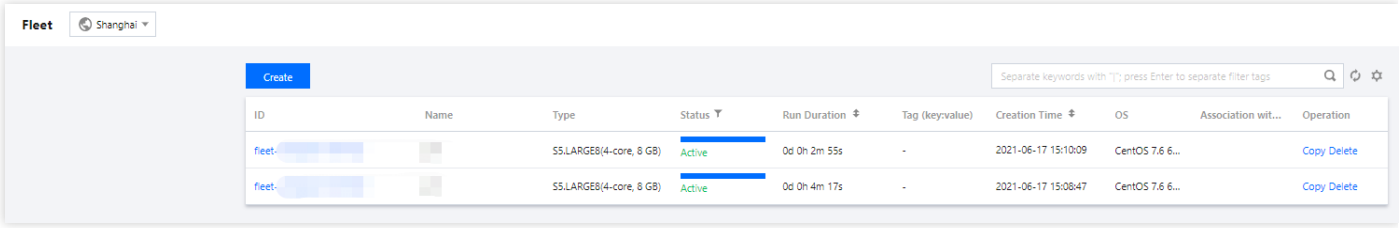
Last updated : 2021-06-28 10:09:36

## Overview

This document describes how to implement zero downtime update through an alias.

## Prerequisites

- Create two server fleets in Shanghai region as instructed below:
  - Complete the first three steps in Demo: click **Quick Upload of Demo Package**, **Quick Creation of Server Fleet**, and **Create Game Server Session** and then click **Complete**.
- You have created server fleet 1 and server fleet 2 (Shanghai region).



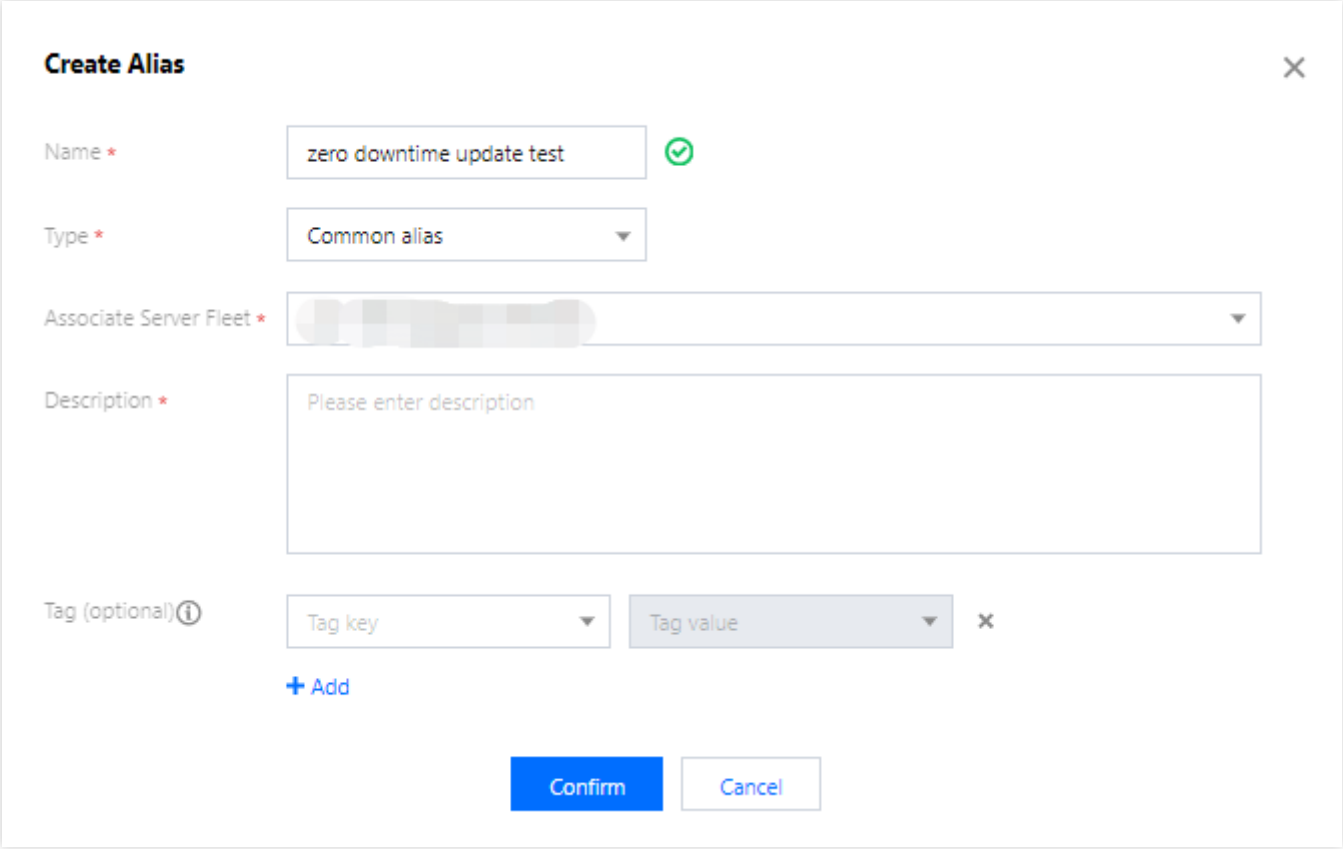

## Directions

### Creating an alias

1. Log in to the GSE console and click **Alias** on the left sidebar.
2. Select the service region in the top-left corner and click **Create**.
3. In the alias creation page, enter the name and description, select the alias type and associated server fleet in the corresponding drop-down list, and click **Confirm**.
   - Name: enter the name of the alias for easy identification in the directory, which is "zero downtime update test" in this example.
   - Type: select common alias or terminal alias, which is "common alias" in this example.
     - Common alias: it points to a fleet, under which the system automatically finds servers and assigns them to clients. If you select common alias, you need to associate an available server fleet.

- Terminal alias: it doesn't point to a fleet. You can describe the reason why the alias cannot be used in **Termination Info**, which will be sent to clients.
- Associate Server Fleet: after selecting common alias, select "server fleet 1".
- Description: enter a short description of the alias for easy identification, which is "test" in this example.
- Tag (optional): the tag is used to manage resources by category from different dimensions. If the existing tag does not meet your requirements, please go to the Tag console to create new tags.

4. After configuring the settings, click **Confirm** to create the alias.



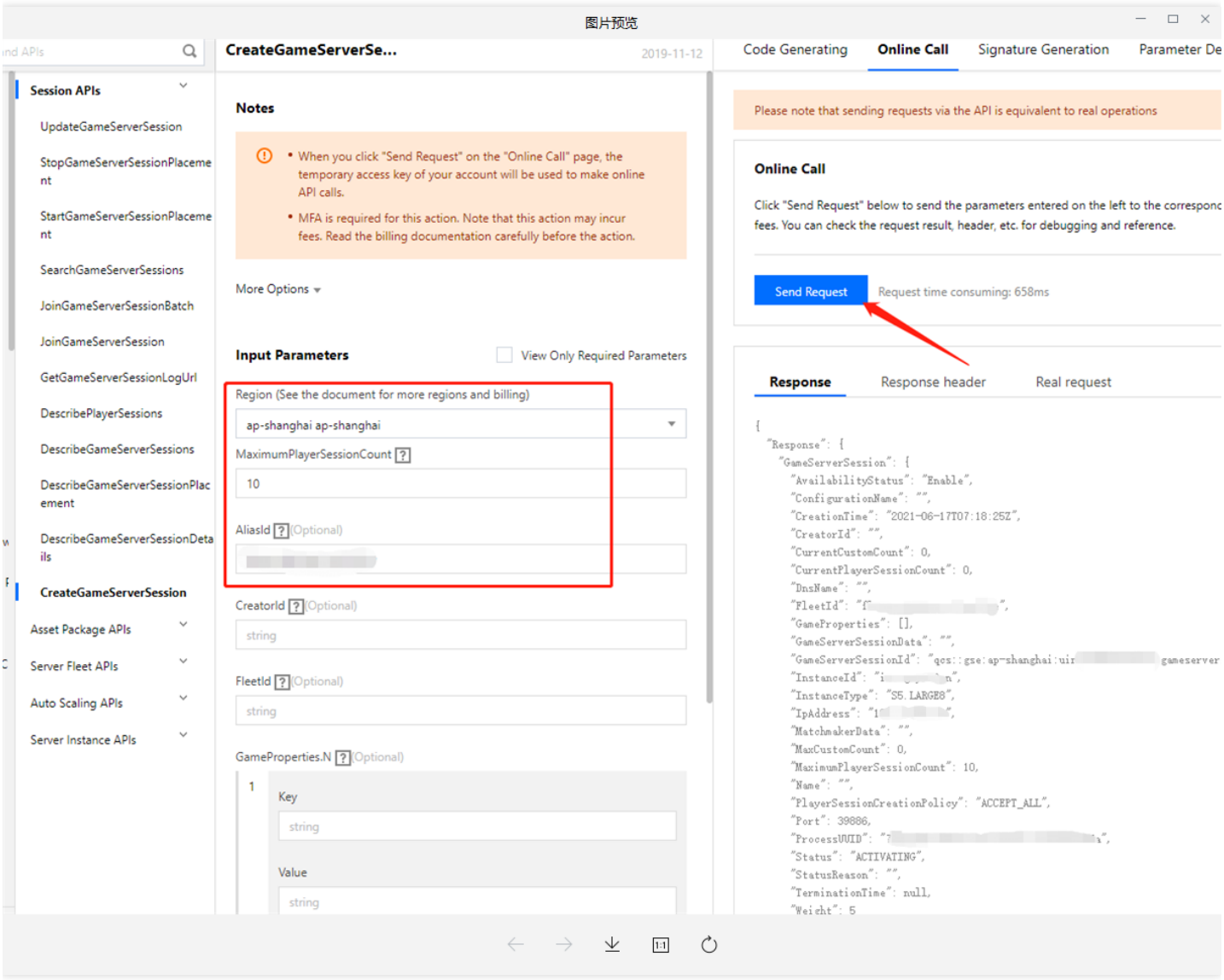

## Creating a game server session

Call the TencentCloud API in the code. This example uses TencentCloud API Explorer for quick creation.

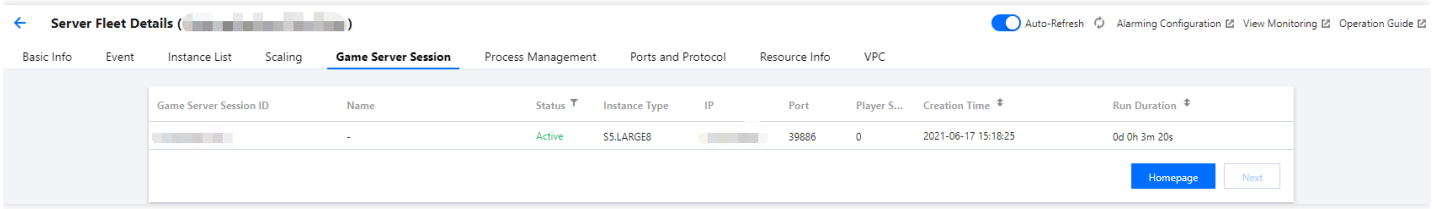
Note：
Input parameter description:

- `Region`：indicates the region, which is “ap-shanghai” (East China (Shanghai)) in this example.

- `MaximumPlayerSessionCount` : indicates the maximum number of players, which is 10 in this example.
- `AliasId` : indicates the alias ID, which is the ID of the newly created alias in this example.

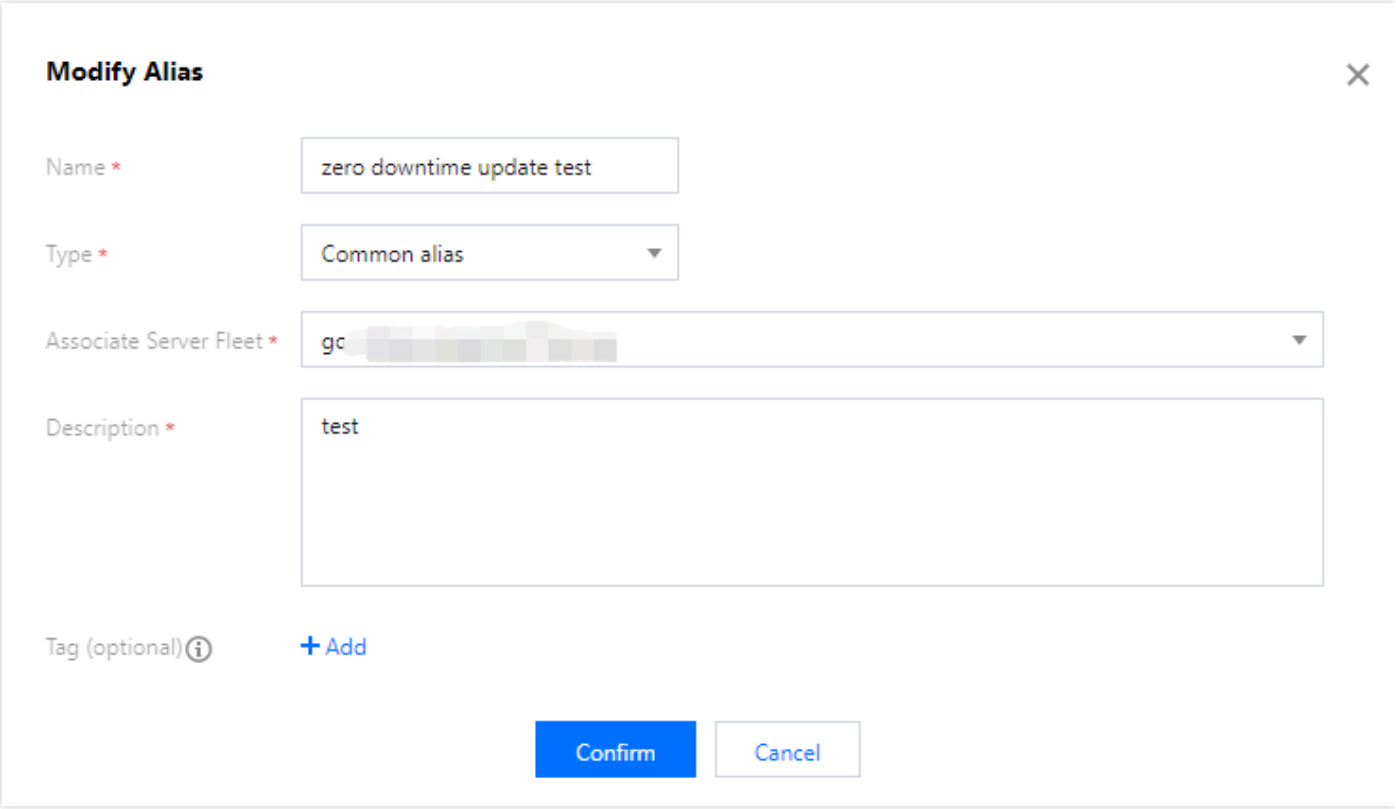



If a game server session is successfully created through TencentCloud API Explorer, you can see that it is generated in server fleet 1.



## Modifying the alias configuration

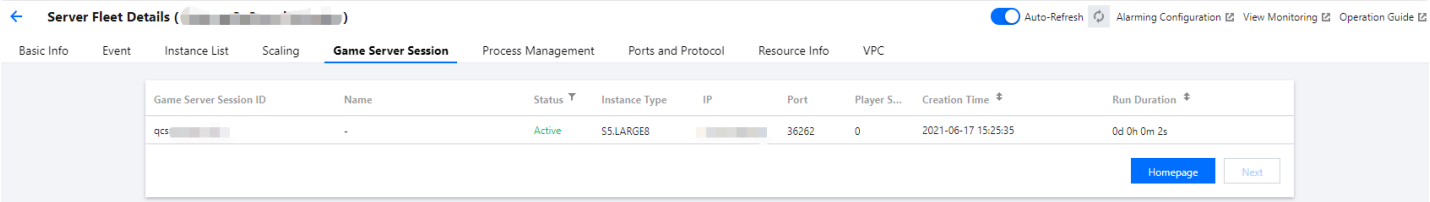
1. In the console, click **Alias** on the left sidebar to enter the alias list page.
2. Select the created alias and click **Modify** to enter the alias editing page, modify the alias configuration, and set "Associate Server Fleet" to "Server fleet 2".




## Creating another game server session

> Note :
> Create another game server session with the same alias.

Create another game server session through TencentCloud API Explorer, and you can see that a game server session is generated in, that is, assigned to server fleet 2.



# Notes on Zero Downtime Updates

For version updates, you can create a new server fleet and point the alias to it.

- Automatic reduction will be performed on the old server fleet as the game server sessions decrease.
- Automatic expansion will be performed on the new server fleet as the game server sessions increase, thus implementing zero downtime update.

# Nearby Resource Scheduling

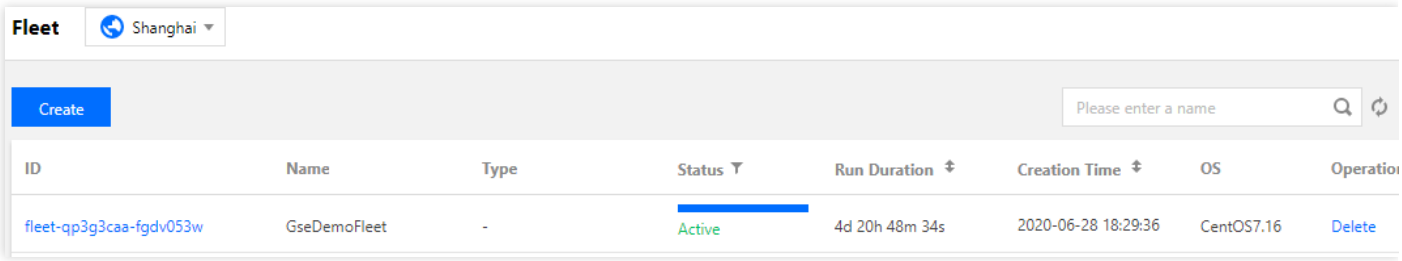
Last updated : 2021-05-20 10:18:11

## Overview

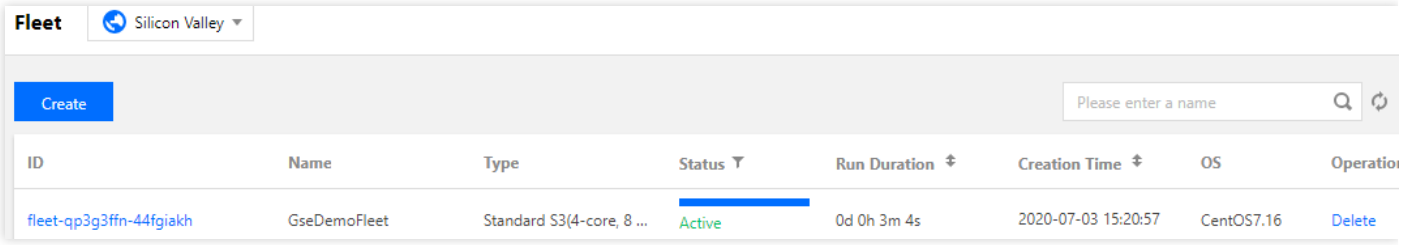
This document describes how to implement nearby resource scheduling through a game server queue.

## Prerequisites

- Create two server fleets in Shanghai and Silicon Valley regions as instructed below:
  - Complete the first three steps in the Demo: click **Quick Upload of Demo Package**, **Quick Creation of Server Fleet**, and **Create Game Server Session** and then click **Complete**.
- You have created server fleet 1 (Shanghai region).




- You have created server fleet 2 (US region).

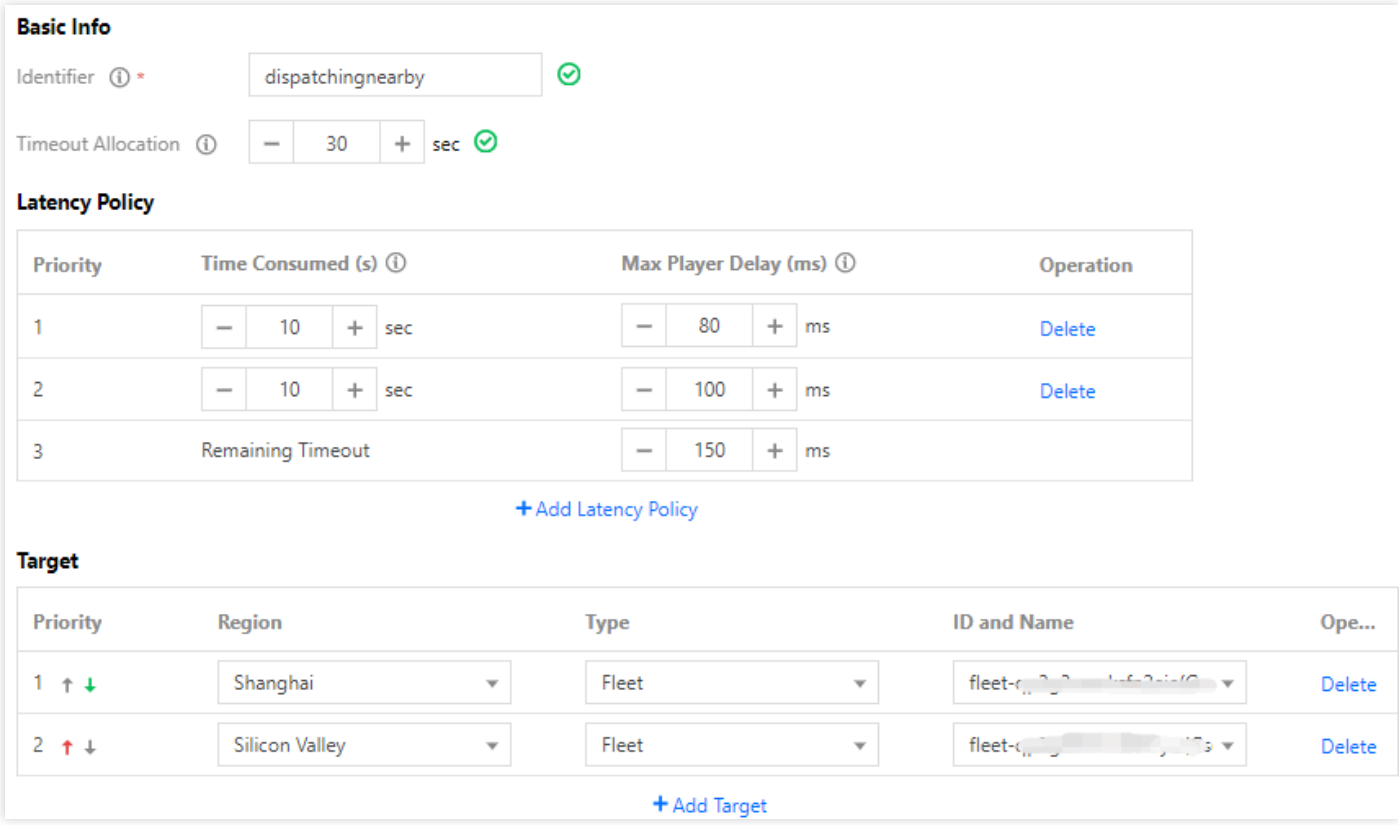



## Directions

### Creating game server queue

1. Log in to the GSE Console and click **Queue** on the left sidebar to enter the game server queue page.

2. Select the service region in the top-left corner and click **Create**.
3. In the game server queue creation page, enter the basic information:
   - Identifier: enter a valid identifier, which can contain letters only and is "dispatchingnearby" in this example.
   - Timeout Allocation: enter the max time that a game server session request can be retained in a multi-region deployment. It can be up to 600 seconds and is 30 seconds in this example.
4. Enter the latency policies:
   - In the first 10s, server fleets whose latency for any players is up to 80 ms are matched and waited for first.
   - In the subsequent 10s (i.e., in the first 20 seconds), server fleets whose latency for any players is up to 100 ms are matched and waited for first.
   - In the last 10s (= 30s - 10s - 10s) of the timeout period, server fleets whose latency for any players is up to 150 ms are matched and waited for.
5. Select the created server fleet 1 (Shanghai region) and server fleet 2 (US region) as the target.
6. Click **OK** to complete creating the game server queue.

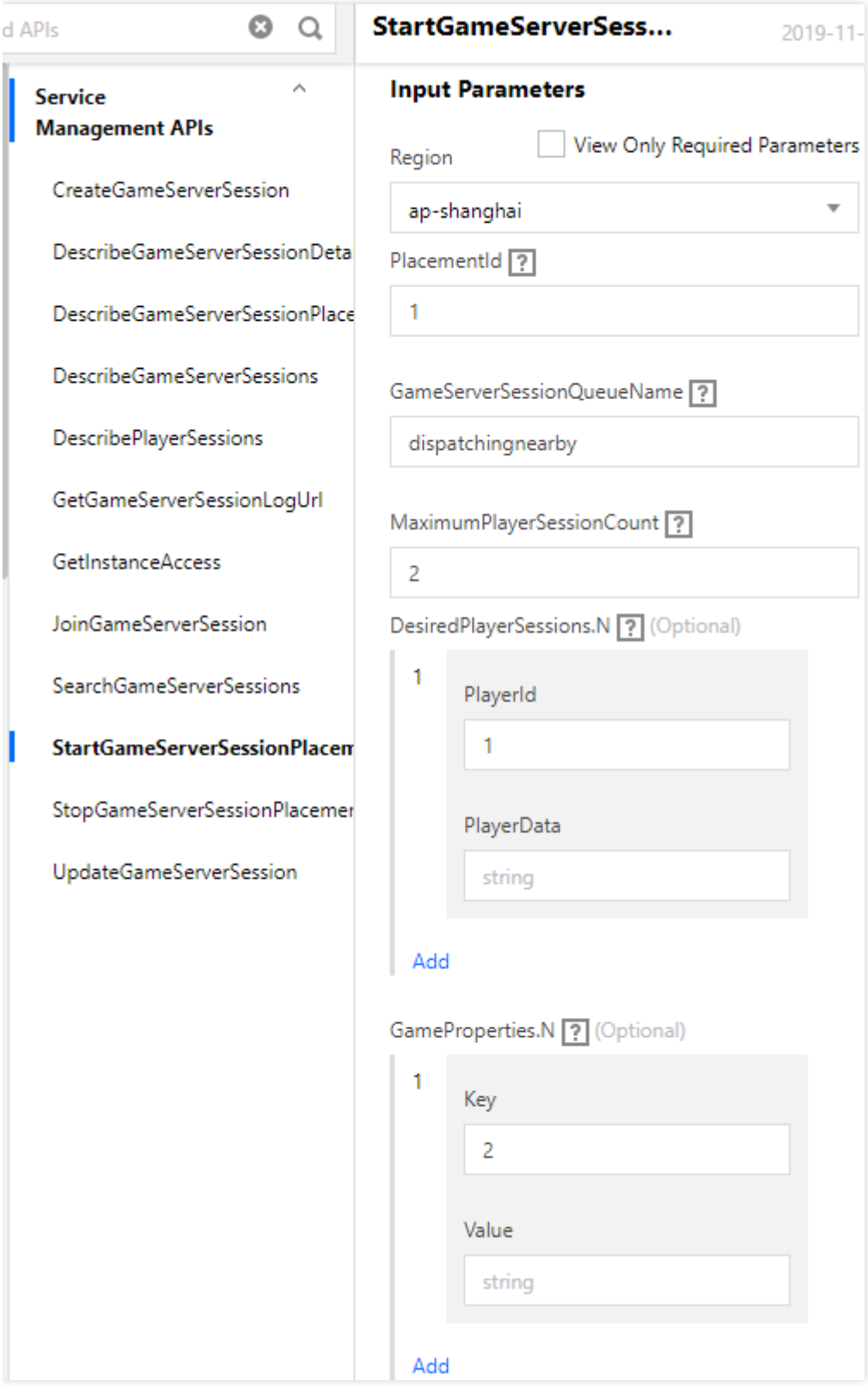


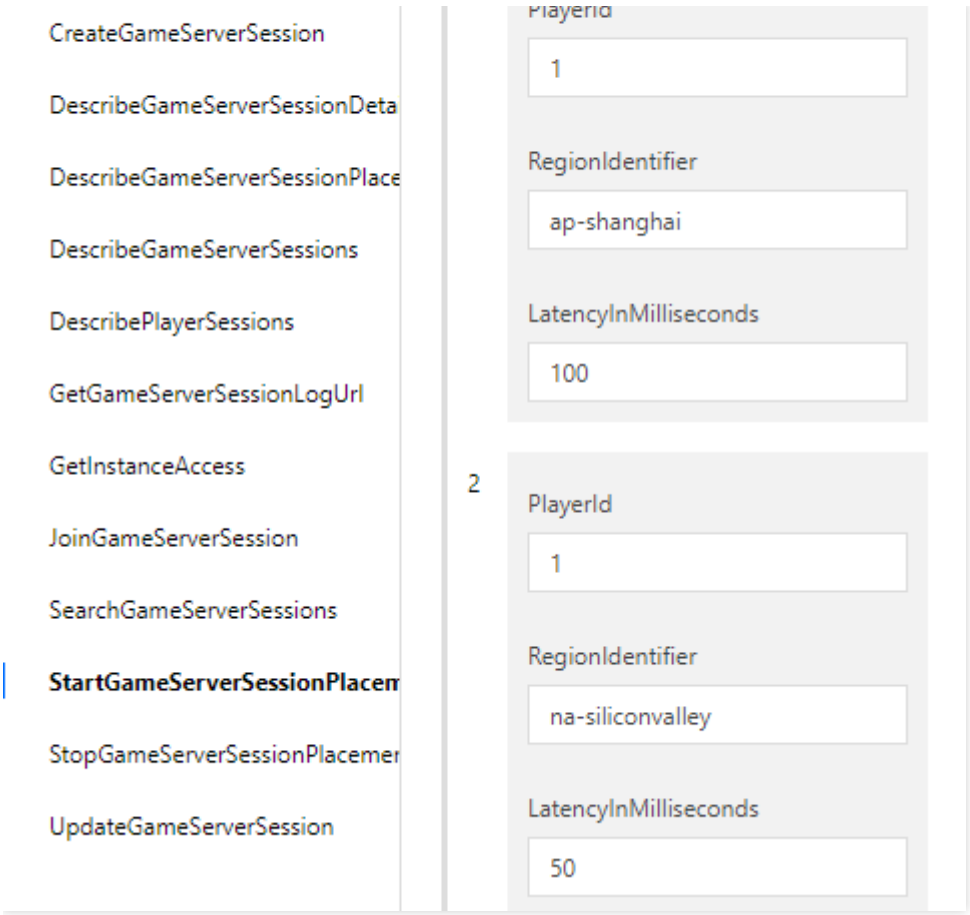

## Starting placing game server session with queue

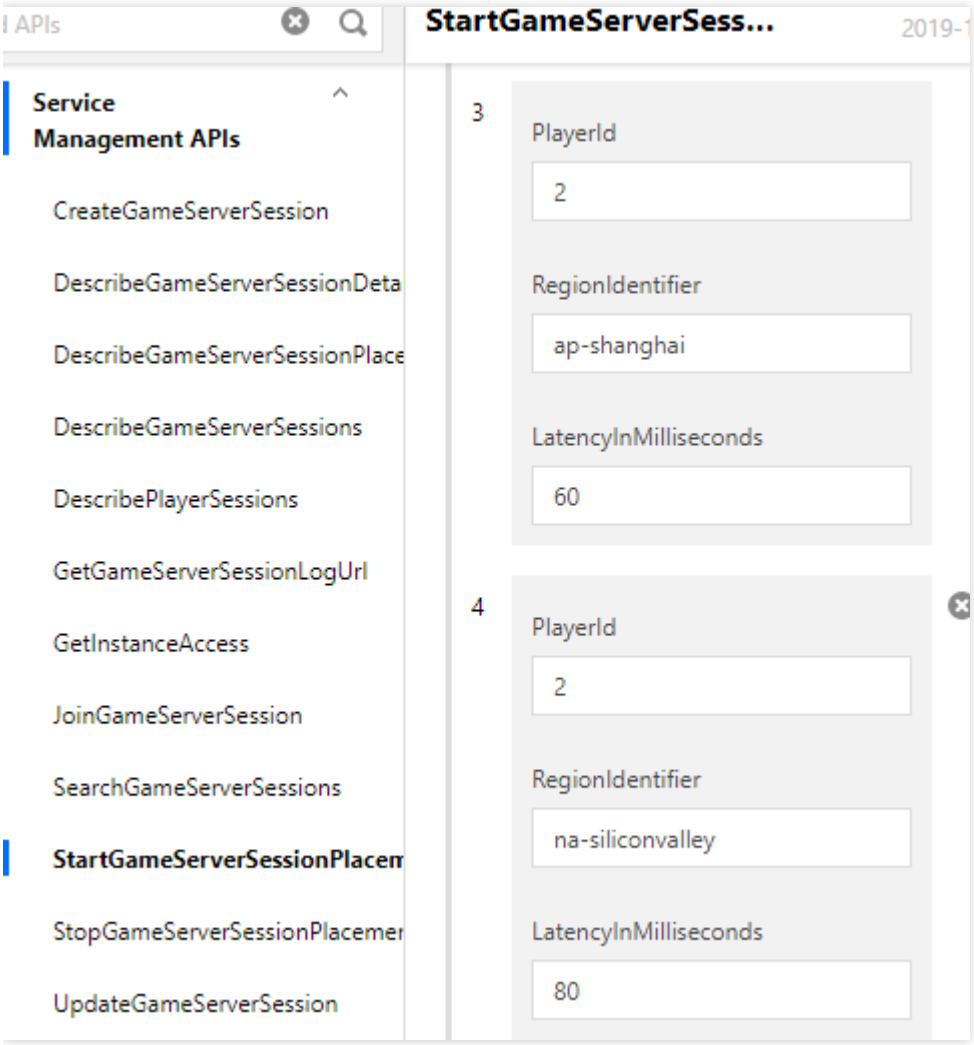
Call the `StartGameServerSessionPlacement` TencentCloud API in the code to place the game server session in the server fleet process. This example uses TencentCloud API Explorer for quick creation.

说明：
Input parameter description

- `Region` indicates the region, which is “ap-shanghai” (East China (Shanghai)) in this example;
- `PlacementId` indicates the unique ID of the game server session placement, which is 1 in this example;
- `GameServerSessionQueueName` indicates the game server session queue name, which is "dispatchingnearby" in this example;
- `MaximumPlayerSessionCount` indicates the maximum number of concurrent players allowed by the game server to connect to the game session, which is 2 in this example;
- `DesiredPlayerSessions.N` indicates the player game session information, where `PlayerId` is the unique player ID associated with the player session. In this example, two values of 1 and 2 are entered respectively;
- `PlayerLatencies.N` indicates the player latency, where `PlayerId` is the player ID, `RegionIdentifier` is the name of the region where the latency occurs, and `LatencyInMilliseconds` is the latency in milliseconds. In this example, four value sets are entered, i.e., [1, ap-shanghai, 100], [1, na-siliconvalley, 50], [2, ap-shanghai, 60], and [2, na-siliconvalley, 80].






**Scheduling result evaluation of latency policy**

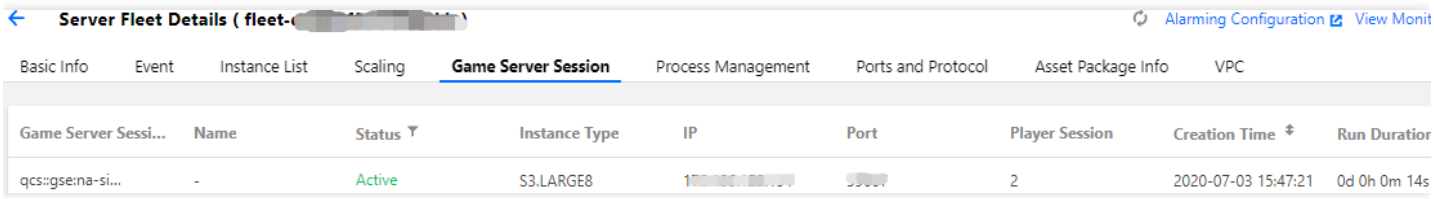
Latency in two players' arrival at the target address:

- The latency for player 1 is 100 ms to Shanghai and 50 ms to Silicon Valley.
- The latency for player 2 is 60 ms to Shanghai and 80 ms to Silicon Valley.

Since the latency policy specifies that in the first 10 seconds, servers in regions where the latency for any players is up to 80 ms will be matched first, the game server session will be scheduled to the Silicon Valley region.

**Test result returned after API call:**

The game server session is scheduled to server fleet 2 (US region).



# Cross-Region Disaster Recovery

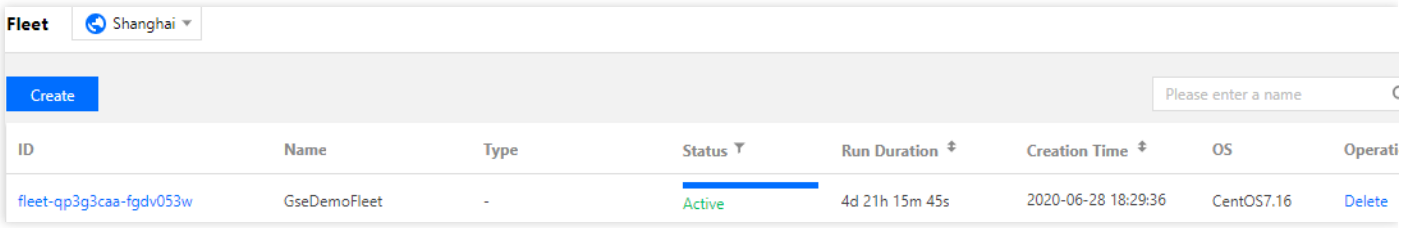
Last updated : 2020-08-04 11:33:32

## Overview

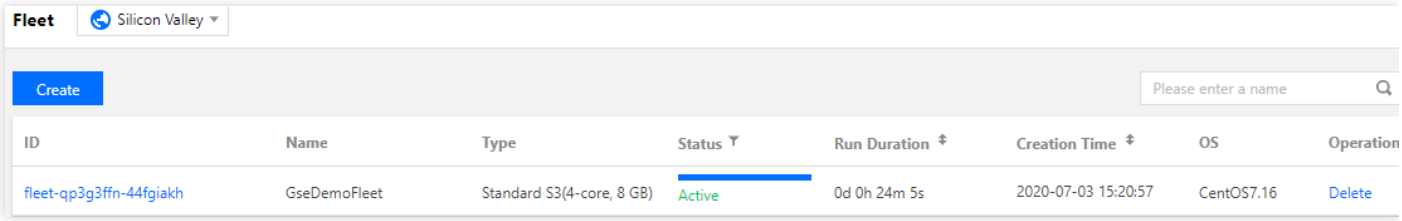
This document describes how to implement cross-region disaster recovery through a game server queue.

## Prerequisites

- Create two server fleets in Shanghai and Silicon Valley regions as instructed below:
  - Complete the first three steps in Demo: click **Quick Upload of Demo Package**, **Quick Creation of Server Fleet**, and **Create Game Server Session** and then click **Complete**.
- You have created server fleet 1 (Shanghai region).




- You have created server fleet 2 (US region).

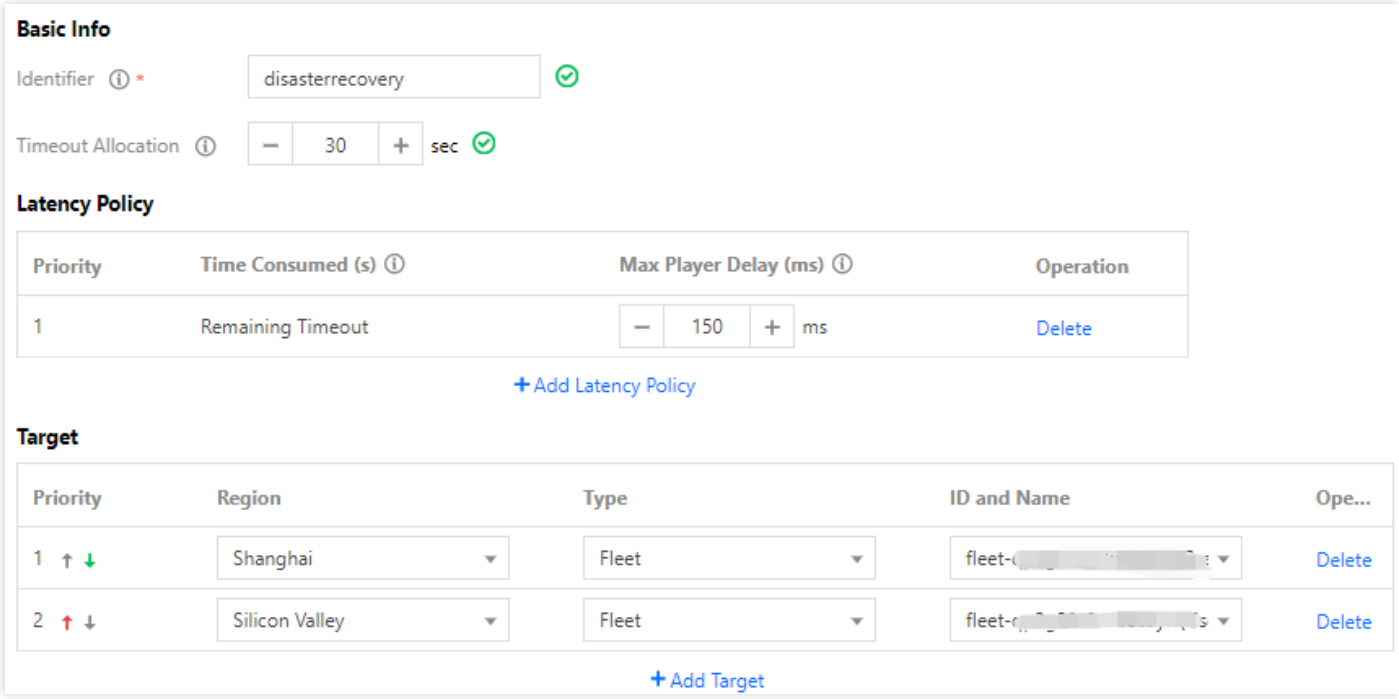



## Directions

### Creating game server queue

1. Log in to the GSE Console and click **Queue** on the left sidebar to enter the game server queue page.
2. Select the service region in the top-left corner and click **Create**.

3. In the game server queue creation page, enter the basic information:
   - Identifier: enter a valid identifier, which can contain letters only and is "disasterrecovery" in this example.
   - Timeout Allocation: enter the max time that a game server session request can be retained in a multi-region deployment. It can be up to 600 seconds and is 30 seconds in this example.
4. Enter the latency policy:
   - Use only one policy where the max player latency is specified as 150 ms to search for server fleets whose latency is up to 150 ms for any player.
5. Select the created server fleet 1 (Shanghai region) and server fleet 2 (US region) as the target.
6. Click **OK** to complete creating the game server queue.




## Requesting a server address as no failure occurs

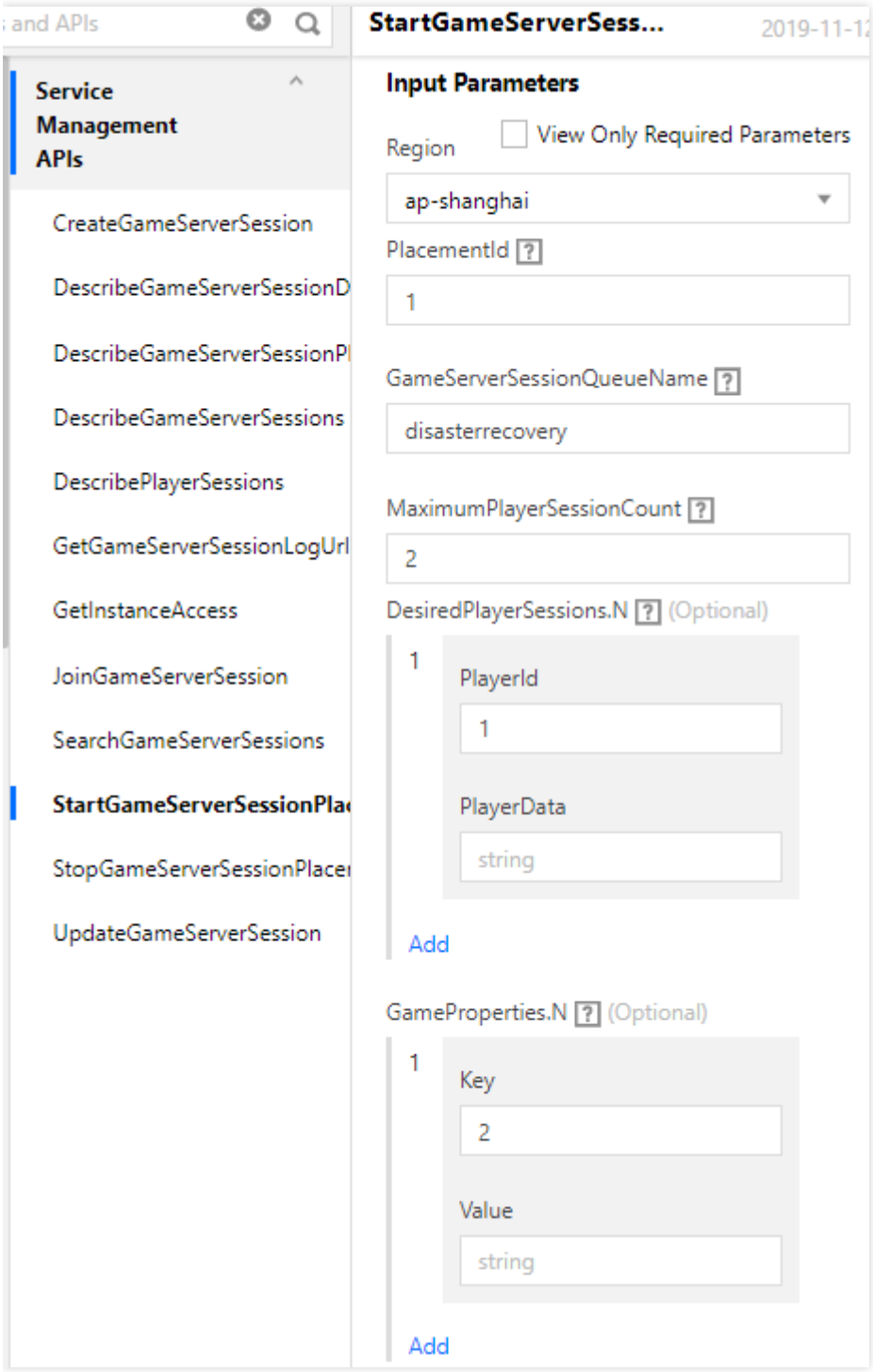
Call the `StartGameServerSessionPlacement` TencentCloud API in the code to place the game server session in the server fleet process. This example uses TencentCloud API Explorer for quick creation.

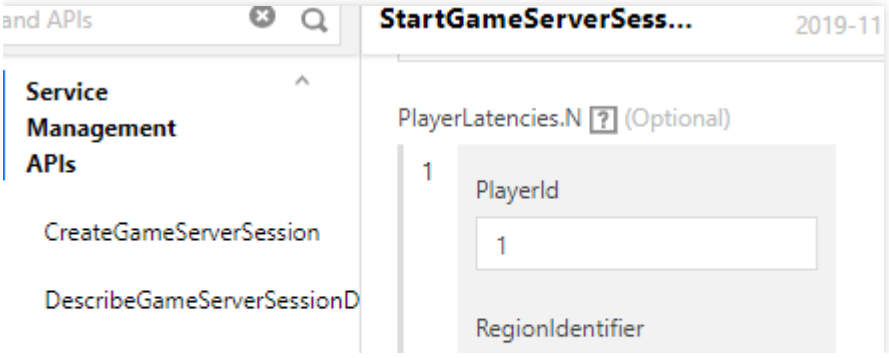
**Note :**

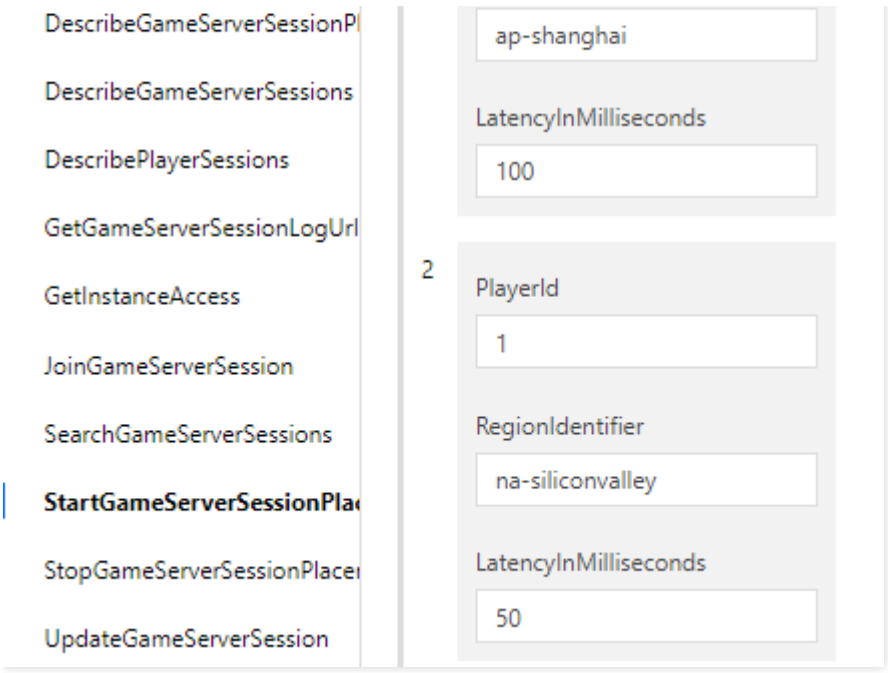
Input parameter description:

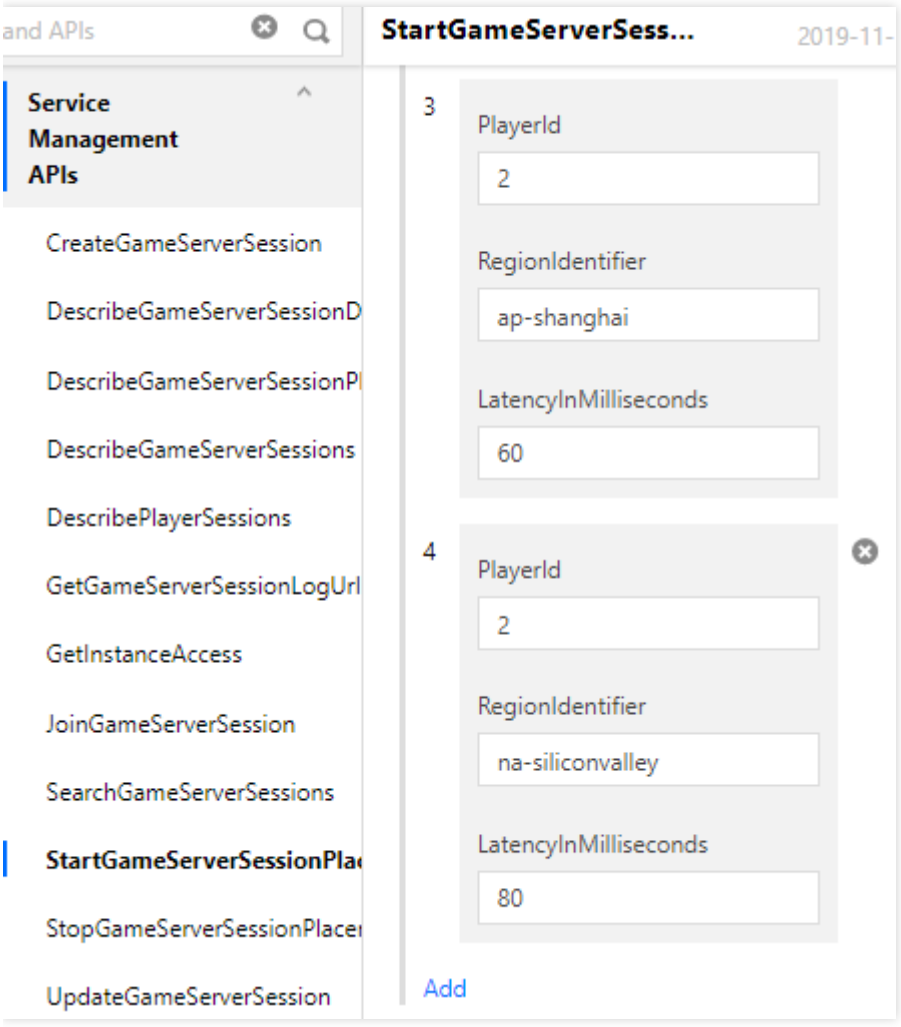
- `Region` indicates the region, which is “ap-shanghai” (East China (Shanghai)) in this example;
- `PlacementId` indicates the unique ID of the game server session placement, which is 1 in this example;

- `GameServerSessionQueueName` indicates the game server session queue name, which is "disasterrecovery" in this example;
- `MaximumPlayerSessionCount` indicates the maximum number of concurrent players allowed by the game server to connect to the game session, which is 2 in this example;
- `DesiredPlayerSessions.N` indicates the player game session information, where `PlayerId` is the unique player ID associated with the player session. In this example, two values of 1 and 2 are entered respectively;
- `PlayerLatencies.N` indicates the player latency, where `PlayerId` is the player ID, `RegionIdentifier` is the name of the region where the latency occurs, and `LatencyInMilliseconds` is the latency in milliseconds. In this example, four value sets are entered, i.e., [1, ap-shanghai, 100], [1, na-siliconvalley, 50], [2, ap-shanghai, 60], and [2, na-siliconvalley, 80].



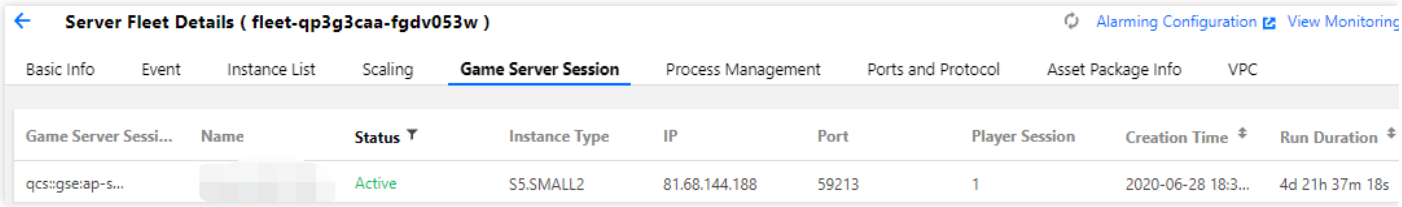



**Scheduling result evaluation of latency policy:**

Latency in two players' arrival at the target address:

- The latency for player 1 is 100 ms to Shanghai and 50 ms to Silicon Valley.
- The latency for player 2 is 60 ms to Shanghai and 80 ms to Silicon Valley.
  As the latency policy specifies that only servers whose latency for any player is up to 150 ms can be matched and both the Silicon Valley and Shanghai regions meet the requirement, a game server session will be automatically created in server fleet 1 (Shanghai region) with a higher priority.



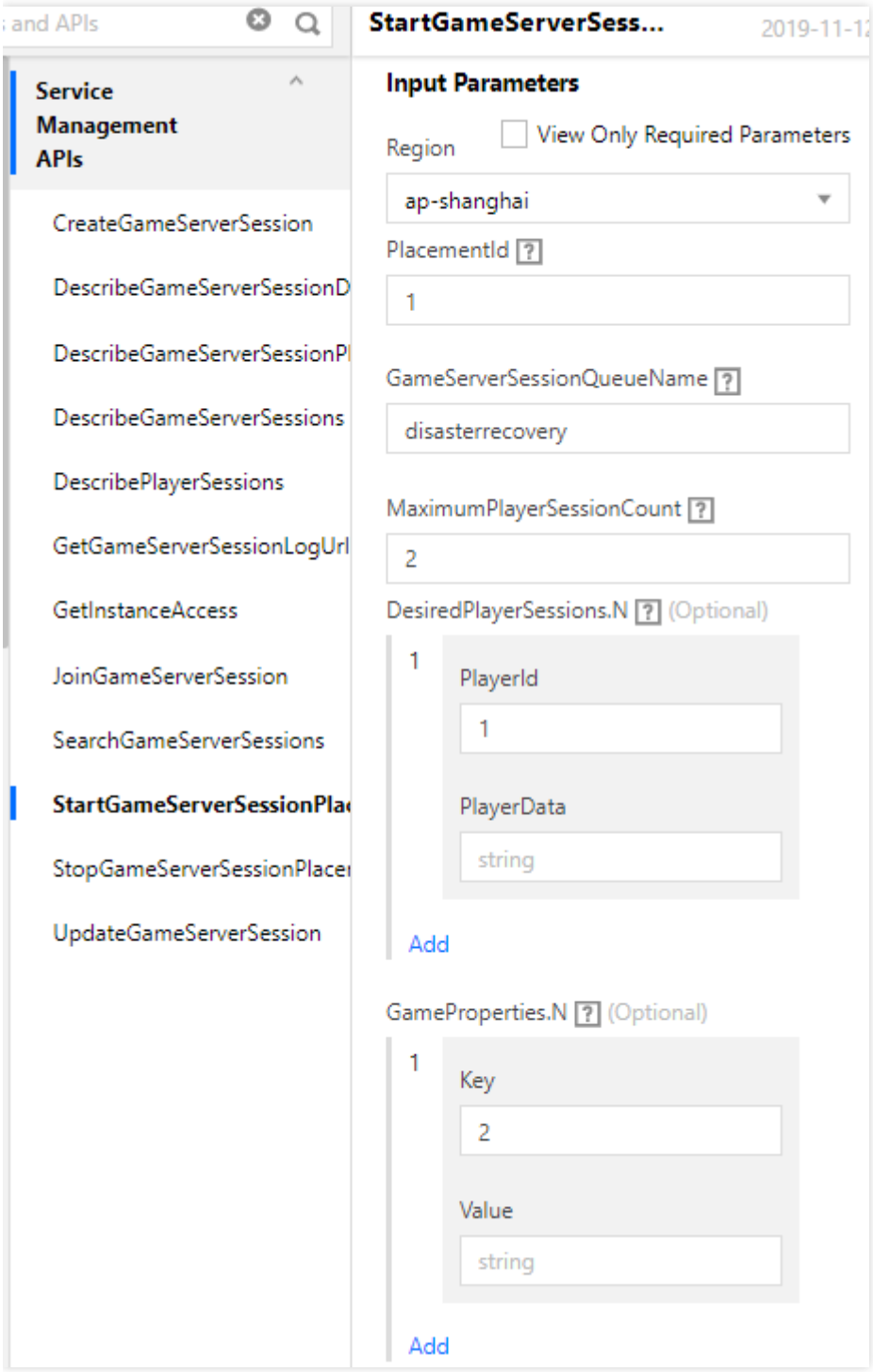

## Automatic disaster recovery in case of failure

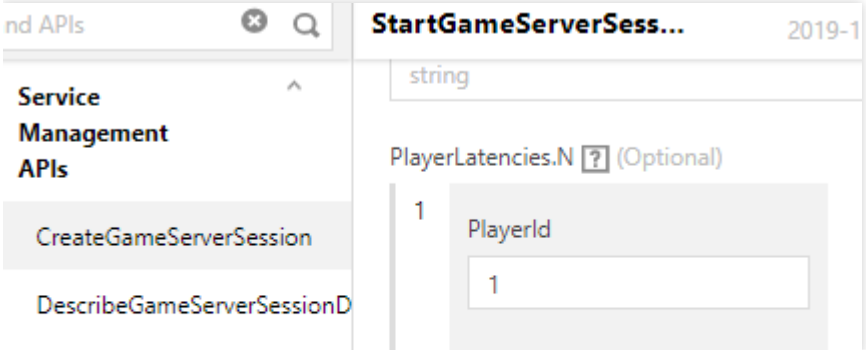
Suppose the Shanghai region fails and its speed cannot be tested.

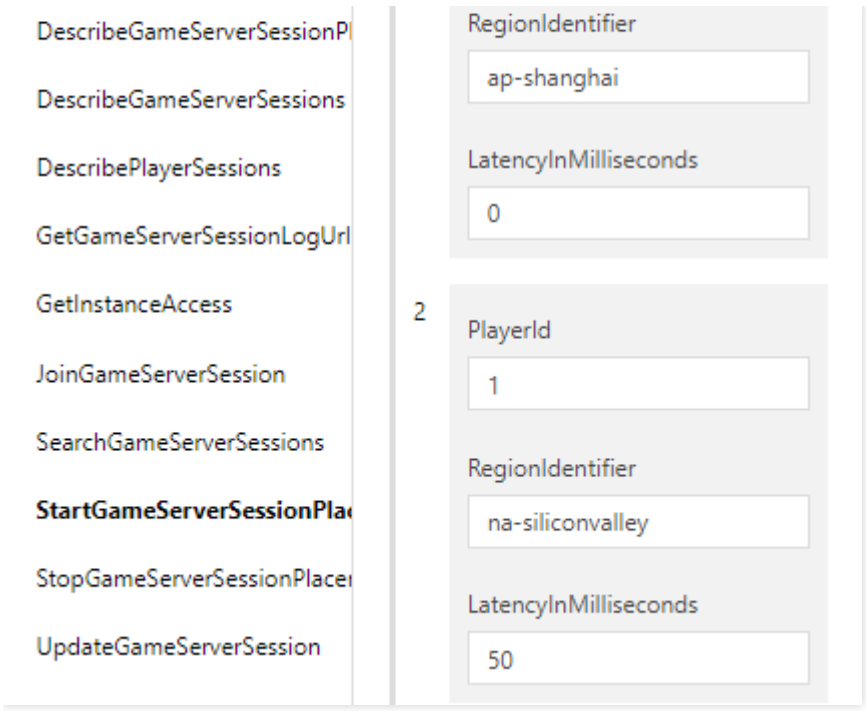
**Note :**

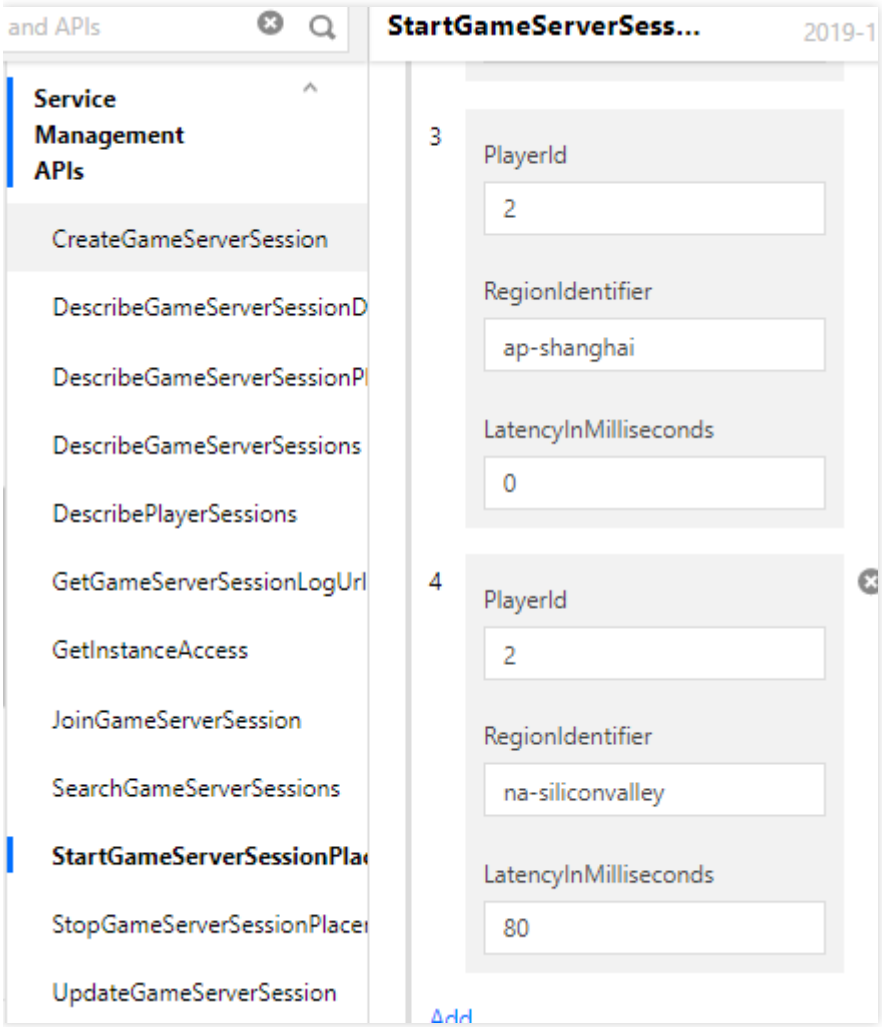
Input parameter description:

- `PlayerLatencies.N` indicates the player latency, where `PlayerId` is the player ID, `RegionIdentifier` is the name of the region where the latency occurs, and `LatencyInMilliseconds` is the latency in milliseconds. In this example, four value sets are entered, i.e., [1, ap-shanghai, 0], [1, na-siliconvalley, 50], [2, ap-shanghai, 0], and [2, na-siliconvalley, 80]. In case that the region speed cannot be tested, enter 0 or an infinite number for the latency value, or leave it empty. In this example, the latency to Shanghai is entered as 0.
- Keep the rest parameters the same as the ones above.





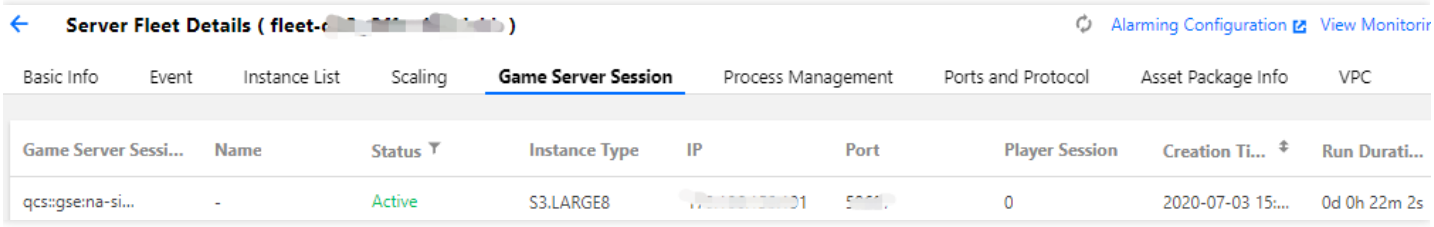



**Scheduling result evaluation of latency policy:**

Latency in two players' arrival at the target address:

- The latency for player 1 is 0 ms to Shanghai and 50 ms to Silicon Valley.
- The latency for player 2 is 0 ms to Shanghai and 80 ms to Silicon Valley.

A latency of 0 ms to the Shanghai region indicates that the latency cannot be measured due to a failure in Shanghai; therefore, a game server session will be automatically created in server fleet 2 in the US region.




## Manual disaster recovery in case of failure

If a region fails, you need to manually remove server fleets in it from the target list in the game server queue, and GSE will schedule game server sessions to the remaining server fleets in the target list.