

Game Server Elastic-scaling

Product Introduction

Product Documentation



Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Product Introduction

Overview

Strengths

How It Works

Use Cases

Product Introduction

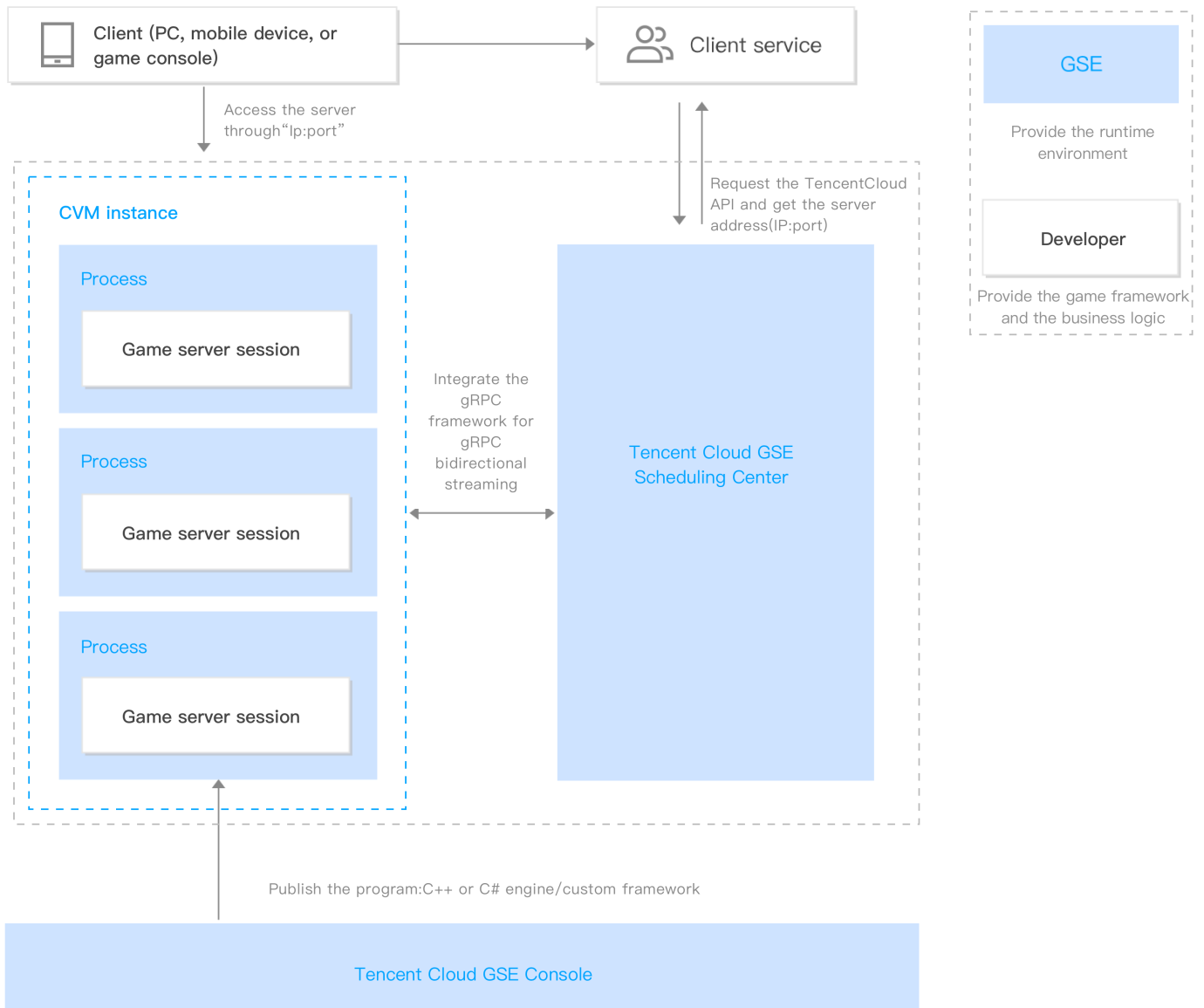
Overview

Last updated : 2021-04-16 14:32:28

Tencent Cloud Game Server Elastic-scaling (GSE) provides dedicated game server hosting services for the deployment and scaling of stateful games. It supports service discovery, flexible server scaling, and optimal resource scheduling. GSE helps developers quickly build a stable and low-latency deployment environment for multiplayer games while reducing OPS costs. It can be used to deploy and run Unreal and Unity game engines, as well as server frameworks written in C#, C++ or any language that supports gRPC. It is ideal for stateful games that need to remember data such as battle servers and push notifications in FPS, MOBA, turn-based games, MMORPG, chess and card games, and more.

Product Architecture

GSE provides a runtime environment for game code packages where you can perform service discovery, scaling, cross-region deployment, and nearby scheduling. You can integrate the GSE ServerSDK into your server framework so that clients can request access to game servers through APIs and GSE will return the optimal game server for client access.



Features

Program deployment and update

Program deployment

You can upload code packages and dependencies through GSE, which will deploy your code to the server fleet and launch it according to the configuration.

Zero downtime update

Generally, servers need to be suspended when games are updated, but GSE implements updates with zero downtime by providing an alias mechanism.

A/B Test

You can easily conduct or end an A/B test with GSE.

Server instance management

Auto scaling

- Auto scaling is performed once every day: games have peak and off-peak hours. Surges generally happen at noon and in the evenings, and demand falls after midnight. With GSE, you can set a selected instance type and a scaling scope, and GSE will auto-scale game servers according to player traffic within the specified scope throughout the day.
- Stateful reduction: GSE will not remove instances with running processes. When the server load drops and reduction is triggered, GSE will inform the game processes that the server is being removed and prevent new game server sessions from being assigned to the server. However, it will not forcibly terminate the instance, so clients can still be connected. The instance will be removed only after the game processes initiate an end command.

Cross-region deployment

GSE supports cross-region deployment, and server fleets built in multiple regions make up a queue. When the queue is requested, the system will automatically select a server fleet in a region for player access. You can also manually adjust the fleet priority. When a region fails, the service will be quickly switched to another region.

Global release

GSE is available in regions such as Shanghai and North America and will soon be available in more regions.

Process management

Process launch

GSE launches a process based on the launch path, launch parameters, and the allowed concurrent processes you specify.

Process preparation

After a process is launched, an API will be called to inform GSE that the process is ready to accept access requests.

Process health check

GSE regularly performs health checks on processes. If a process is found to be unhealthy, it will be blocked and will not be assigned to any caller.

Process end

- If GSE needs to reduce its capacity or if health check fails, it will ask related processes to end. The processes can choose whether to end. If they do not immediately end, GSE will handle them based on the protection policy configured on the console.
- Processes can proactively call an API to inform GSE that they have ended.

Game server session management

GSE manages and assigns game server sessions. From a service perspective, one game server session represents one gaming round (a battle) or a single service. From a backend program perspective, one game server session corresponds to one process, which will be launched by GSE in advance as configured. Generally, one process matches one session. When a client requests a game server session through TencentCloud API, GSE will assign this session to an idle process.

When the game server session starts

When a caller requests a game server session, GSE will assign an idle and healthy process to start the session.

When the game server session ends

When there are no players on a game server session or the session is unhealthy, the session will end itself or be ended. Logs will be saved before the session ends to facilitate troubleshooting.

Nearby assignment of game server session

GSE can select the nearest region to a player based on the network latency.

Server scaling based on the ratio of available game server sessions

You can configure a game server session buffer as an auto scaling condition for a game server fleet. A game server session buffer is the proportion of available game server sessions.

Monitoring and logging

The system provides CVM instance monitoring, game server session monitoring, and operation logs.

Associated feature

For information on the game player matchmaking feature, see [Game Player Matchmaking \(GPM\)](#). GPM provides flexible and powerful matchmaking logic algorithms. You can customize matchmaking rules, and the matched game battles will be automatically put in the GSE game server queue.

Strengths

Last updated : 2020-10-28 16:13:18

Real-time scalability for reduced costs

All games have peak and off-peak hours. With GSE, you can set a selected instance type and a scaling scope, and GSE will auto-scale game servers according to player traffic within the specified scope, enabling you to meet peak demand while eliminating the need to pay for idle servers during off-peak hours.

Stateful scaling mode

GSE will not remove instances where game server sessions are running. When the server load drops and reduction is triggered, GSE will inform the game processes that the server is being removed and prevent new game server sessions from being assigned to said server. However, it will not forcibly terminate the instance, which may cut client connections. The instance will be removed only after the game processes initiate an end command.

Health check for guaranteed service stability

- Health check is performed on servers, and the runtime environment is monitored in real time. If a server fails, it will be deallocated, and traffic will be rescheduled to other servers in a matter of seconds.
- No manual OPS is required. If a massive failure occurs, the region can be automatically switched based on the speed test result. You can also manually remove the faulty region.

Cross-region deployment for effective disaster recovery

GSE supports cross-region deployment, where server fleets built in multiple regions make up a queue. When the queue is requested, the system will automatically select a server fleet in a region for player access. When a region fails, the service will be quickly switched to another region.

Zero downtime updates for a smooth gaming experience

GSE supports zero downtime updates. A client can request a server in a server fleet through an alias. When the version is updated, you can create a server fleet and point the alias to the new fleet to implement an update with zero downtime.

Global release and nearby access

GSE is available in multiple regions such as Shanghai and North America. It provides a speed test tool to test the latency between a game client and each region and can assign the nearest server fleet to the client based on the test data.

Cross-platform call for easier use

GSE can be called across platforms such as PCs, mobile devices, and game consoles. It supports C++ and C# client engines and custom gaming server frameworks.

Pay-as-you-go billing for lower costs

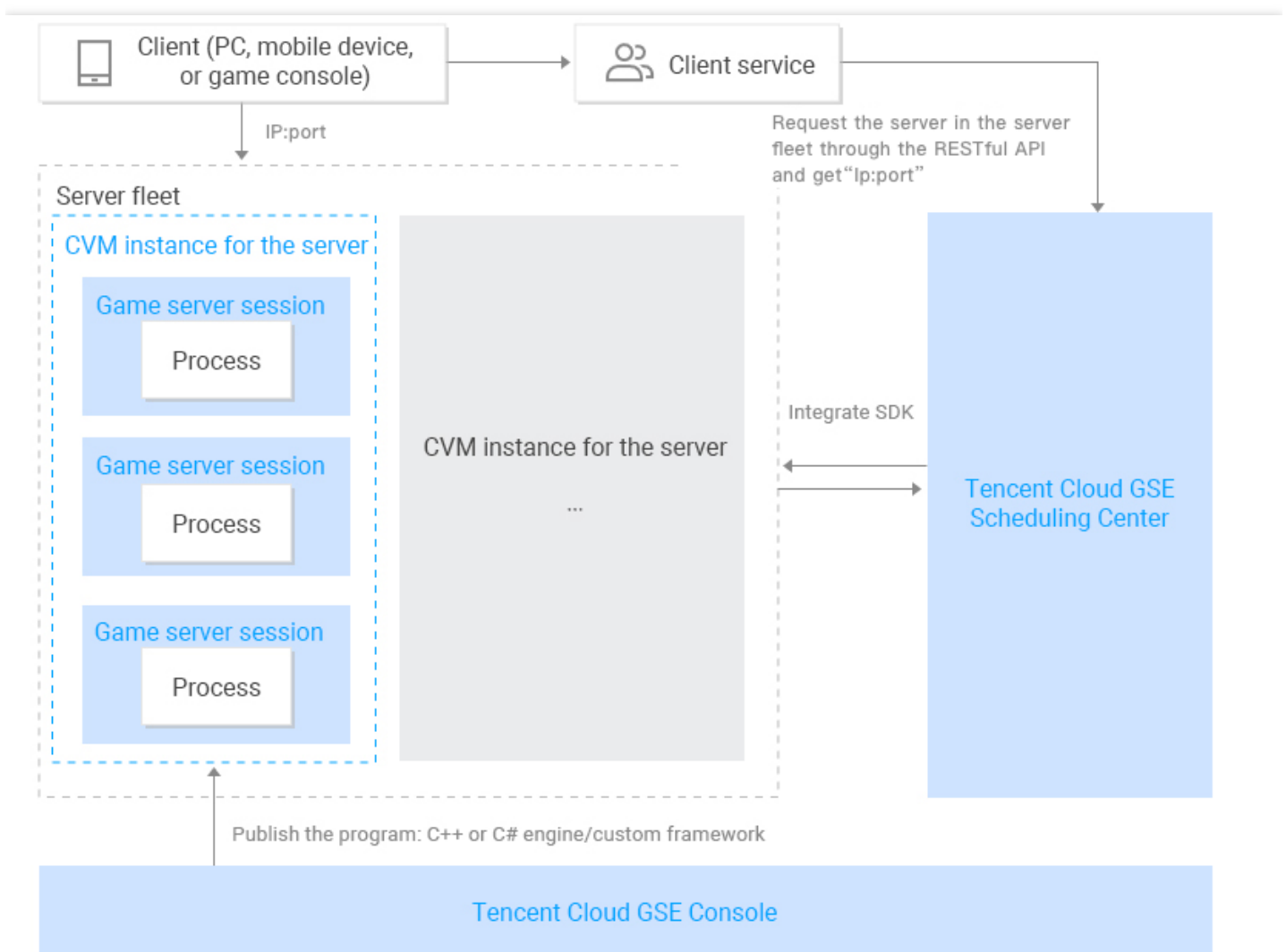
You can set the server instance type and scaling scope in GSE, and instances will be scaled within this scope. Since access traffic to a game has peak and off-peak hours, GSE can perform the auto scaling of servers based on the player access request volume to prepare your business for access peaks and eliminate your need to pay for idle servers during off-peak hours.

How It Works

Last updated : 2020-08-25 09:57:44

Server Fleet

A server fleet consists of multiple CVM instances, each of which runs multiple game server sessions. GSE can manage and assign game server sessions. In general, each session is a process.



CVM auto scaling

- Auto scaling based on game server session buffer
You can configure the game server session buffer, i.e., the ratio of reserved idle server processes,

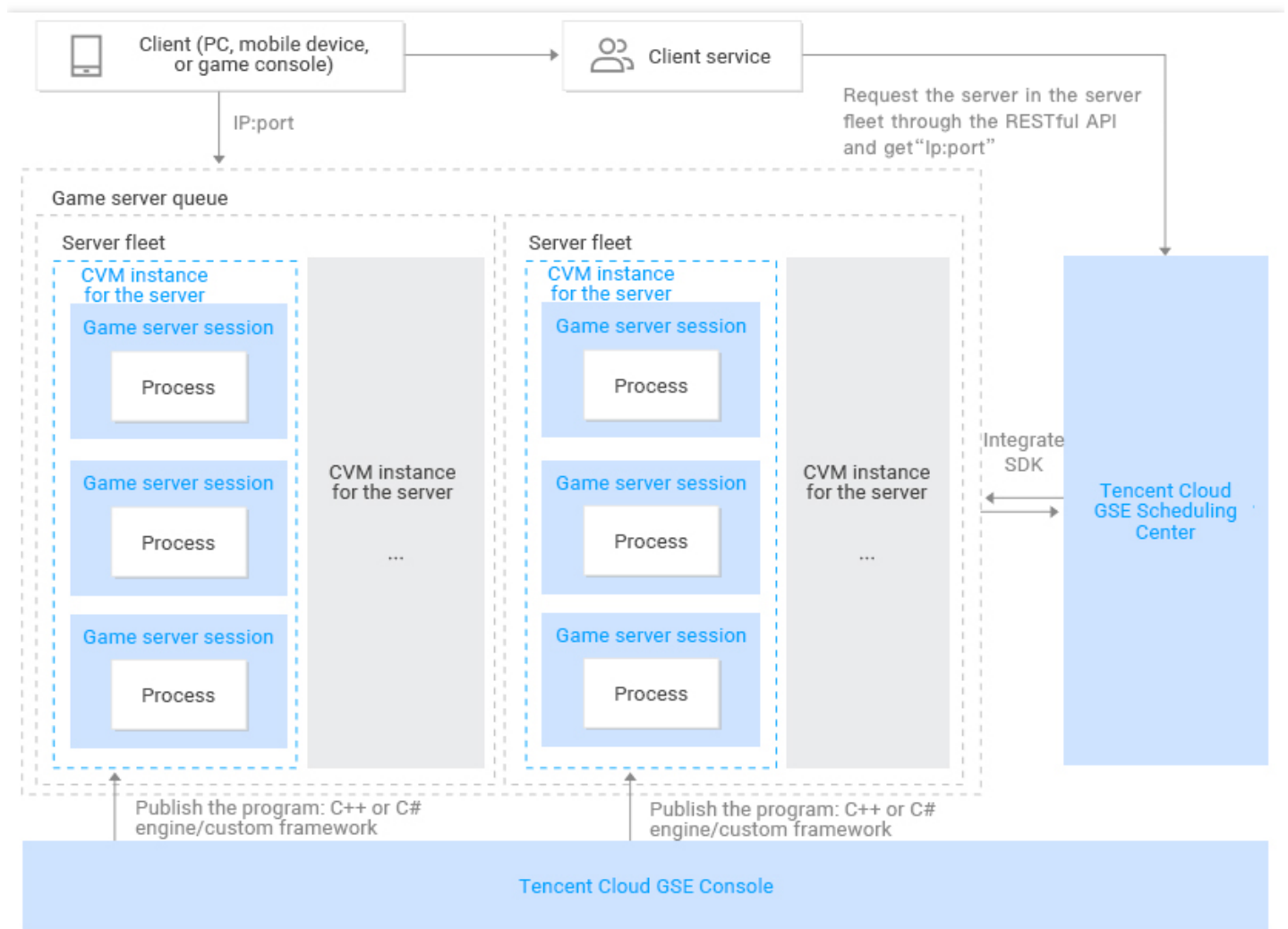
to expand when the idle processes are insufficient, and to reduce when they are sufficient.

- Stateful reduction

GSE will not remove instances where game server sessions are running. During reduction, GSE will notify the game server sessions, which can determine whether or not to reduce immediately. If you selected full protection on the product configuration page and the game server sessions keep running and rejecting reduction, reduction will never be performed.

Game Server Queue

A game server queue is actually a queue of server fleets with configured priorities. It can include server fleets around the world to implement nearby scheduling and disaster recovery.



How disaster recovery works

GSE supports cross-region deployment, in which server fleets built in multiple regions make up a queue. When a region fails, the latency of requests from clients to servers in the region will become very high. In this case, the system will automatically switch to a region with low latency to implement disaster recovery. You can also manually adjust the fleet priority to delete faulty regions from the queue.

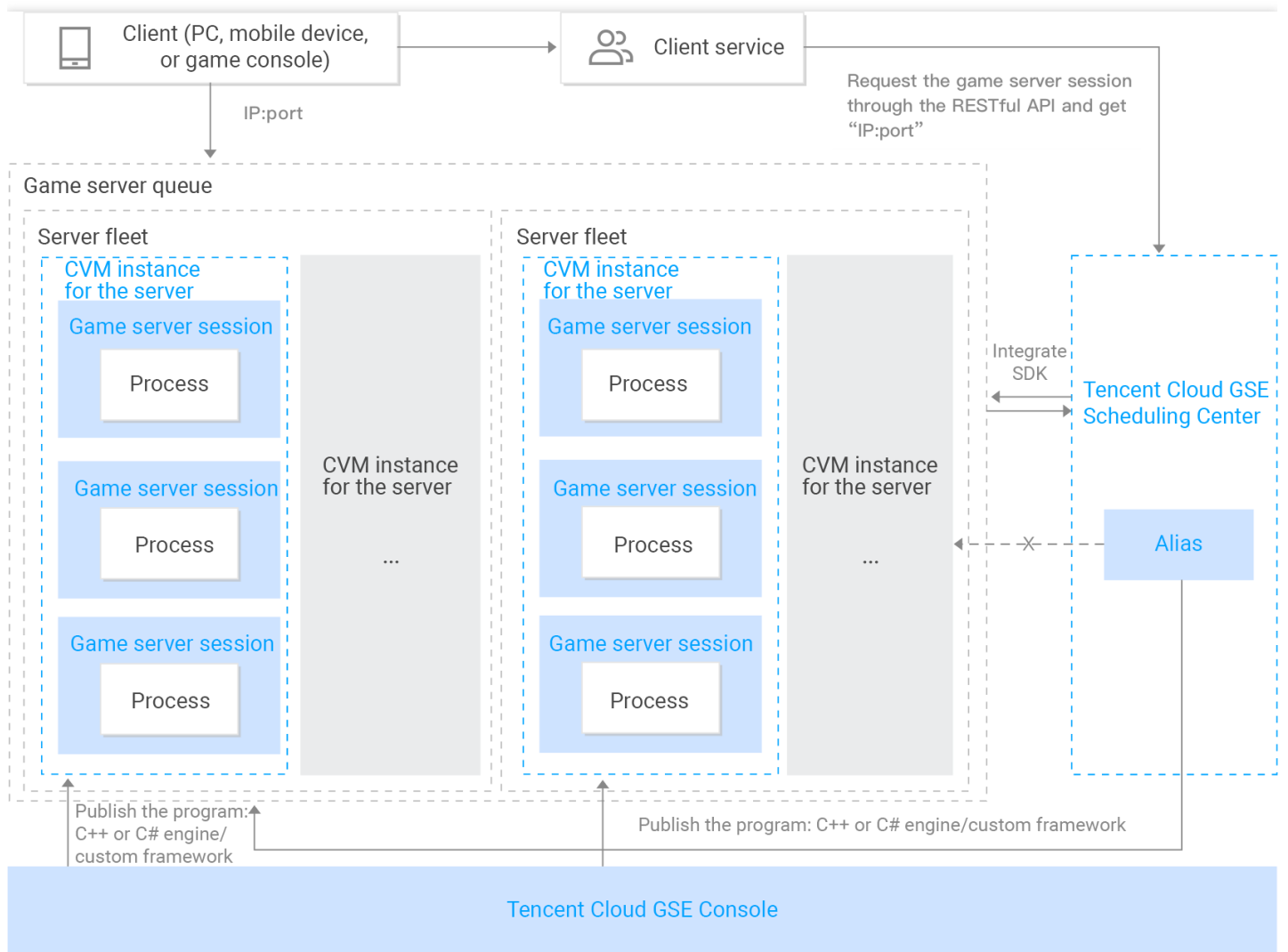
How nearby assignment works

GSE provides a speed test tool that you can use to test the latency of requests from a client to a server. In battle games, GSE will assign a server in the region nearest to all players in the room as a whole.

Alias

A client can request a server in a server fleet through an alias. When the version is updated, you can create a server fleet and point the alias to the new fleet so that the client can still call the same alias

and implement non-stop update.



Use Cases

Last updated : 2021-04-12 14:22:16

GSE can be applied in different scenarios of different games, as detailed below:

Battle server

On a battle server, a battle usually lasts for several to tens of minutes (no more than one hour), and traffic peaks at noon and in the evenings. During off-peak hours, some servers do not need to be used. Therefore, you can use GSE to add/remove servers instantly during peak/off-peak hours, thus reducing the server cost greatly. In addition, GSE can assign the nearest access region for each battle to guarantee network stability and battle fairness.

GSE is suitable for chess and card games, turn-based/strategy games, and real time battle games. You can create a game server session that represents a room or service where players can battle or chat with each other.

Message push

In commonly used game frameworks, a client needs to sustain a persistent connection with a server so that the server can push messages to the client instantly. Therefore, message push is a core module in many games.

Message push may encounter the following problems during deployment:

- A large number of messages fail to be pushed due to network failures.
- Most messages are pushed by only several highly configured servers. Therefore, a faulty server will have an overly large service impact.

Through GSE, you can implement cross-region disaster recovery with minimal costs. When the server in a region, fails, you can quickly switch to a server in another region. The message push service is distributed among multiple servers so that even if a server fails, the affected scope will be small, and you can quickly switch to another server.