Cloud Application Rendering

Console Guide

Product Documentation

Contents

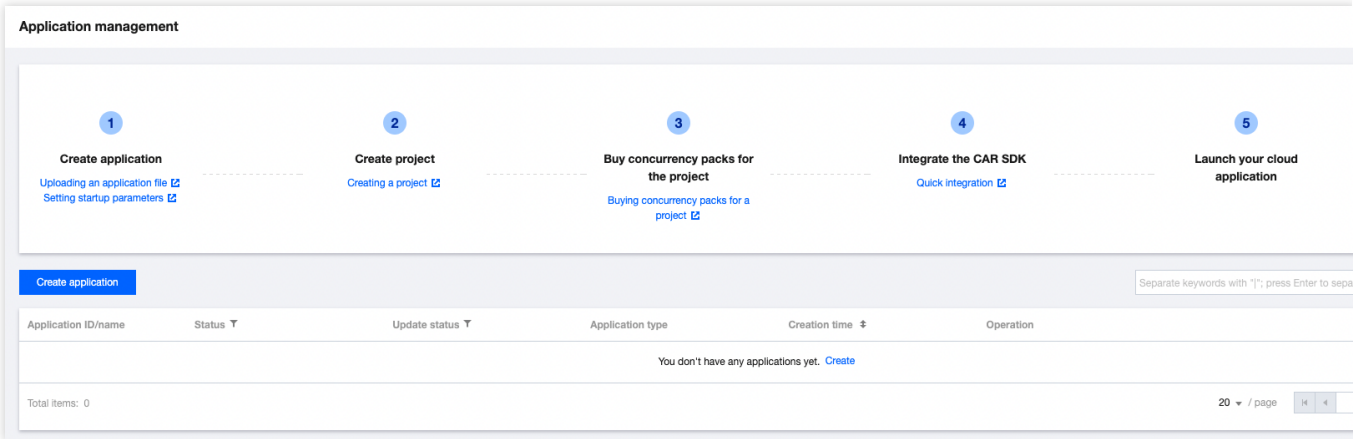
Console Guide
Application Management
Uploading Applications

Last updated：2024-01-26 12:00:38

CAR lets you upload your own applications. You can package your application as a ZIP/RAR/7z file and upload it from your local system or through a URL to deploy it as a cloud application.
CAR supports three application types: cloud 3D, cloud XR, and cloud APK.
Cloud 3D: For general 3D applications (non-AR/VR). These applications are generally controlled using keyboard and mouse, such as virtual event applications and PC games.
Cloud XR (in beta test): For VR/AR/MR applications.
Cloud APK (in beta test): For APK applications.
Note:
Cloud APK and cloud XR are in beta test. For more information, please contact your Tencent Cloud sales rep.

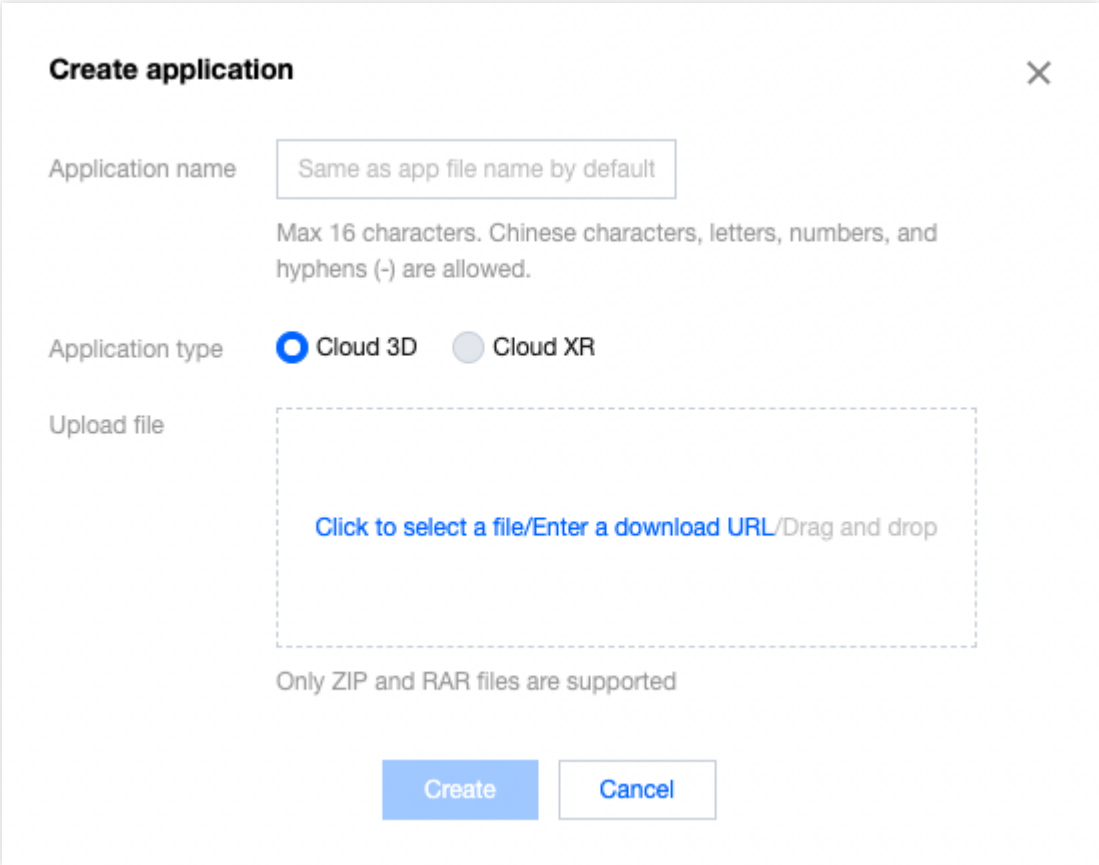
Uploading a cloud 3D application (.exe)

1. Package the application as a ZIP, RAR, or 7z file. Go to the **Applications** page in the CAR console, upload the application file, and wait for the application to be created.
We recommend you use UTF-8 encoding for compression (7-Zip is recommended); otherwise, it may lead to garbled characters or other issues with the files or folder names after decompression.



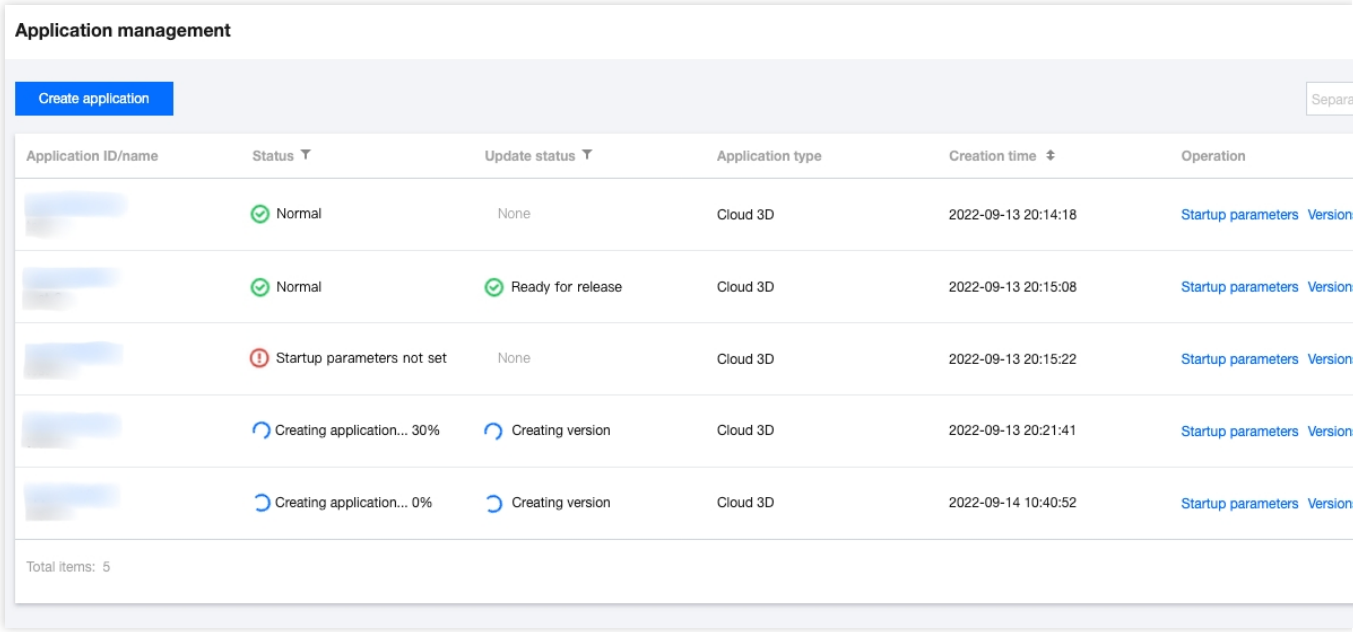
2. In the **Create application** pop-up window, select an application type. You can select a file, drag a file, or enter a URL to upload a file.
Note:

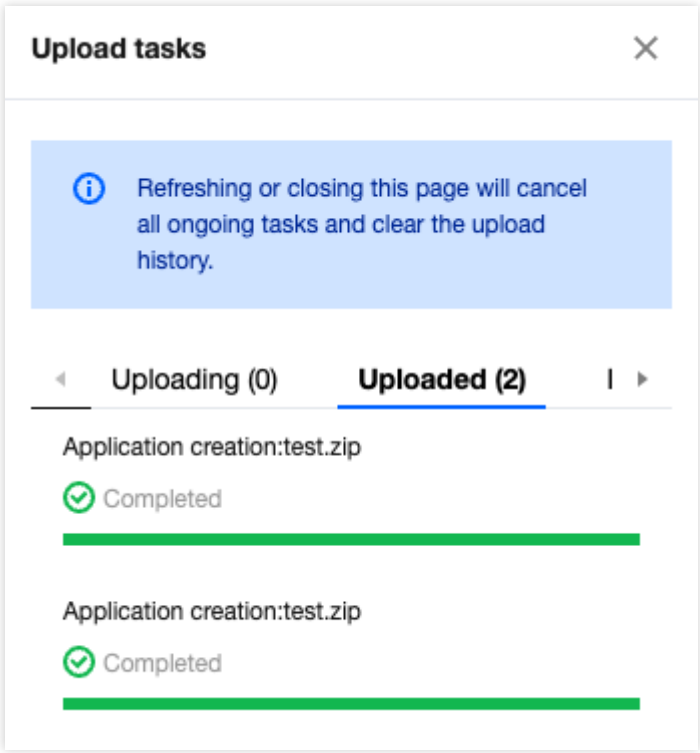
The download URL must be a URL that can be accessed and trigger download directly. Download URLs of cloud disks or other URLs requiring verification are not supported.



3. After you submit the application upload task, you can proceed with the Application Configuration.

4. You can view the application upload progress for a selected or dragged file by clicking **Upload tasks** in the top-right corner. An application uploaded through a URL will be downloaded on the backend. You can directly view the application creation progress in the upload task list on the **Applications** page.





Cloud XR and cloud APK applications

CAR also supports high-precision content in various forms such as VR and AR as well as interaction with VR/AR glasses, mobile phones, tablets, PC, holographic devices, and other devices. This allows for scenarios such as cloud

VR-based multi-person co-training, online large-scale VR entertainment, VR live streaming, and cloud VR/AR exhibitions.

In addition to EXE applications, CAR also supports running APK applications in the cloud so as to enable playing download-free mobile games in advertising, webpages, and other scenarios or to reduce the application file size.

Note:

Cloud XR and cloud APK are in beta test. For more information, please contact your Tencent Cloud sales rep.

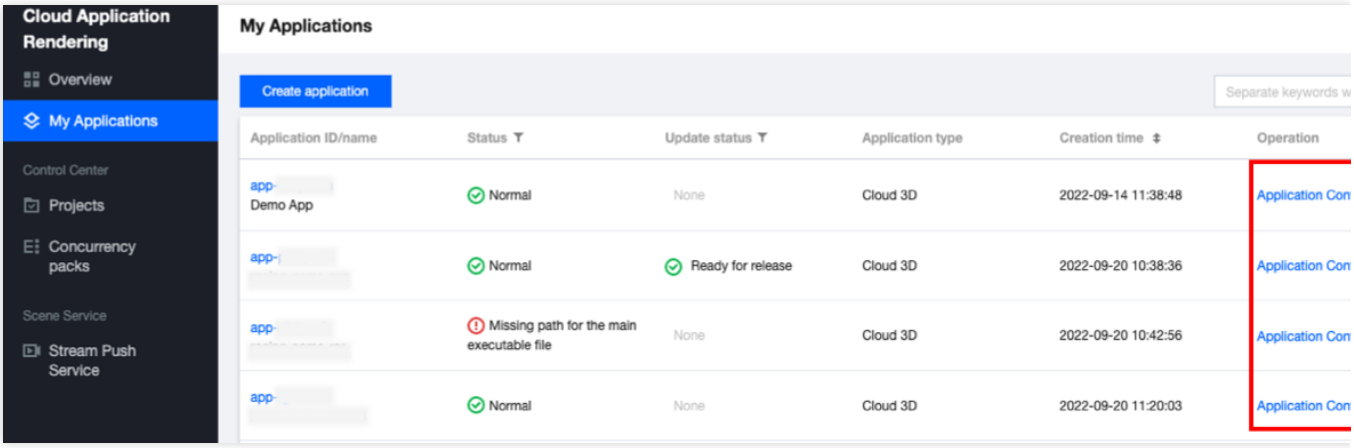
Application Configuration

Last updated：2024-01-26 12:00:38

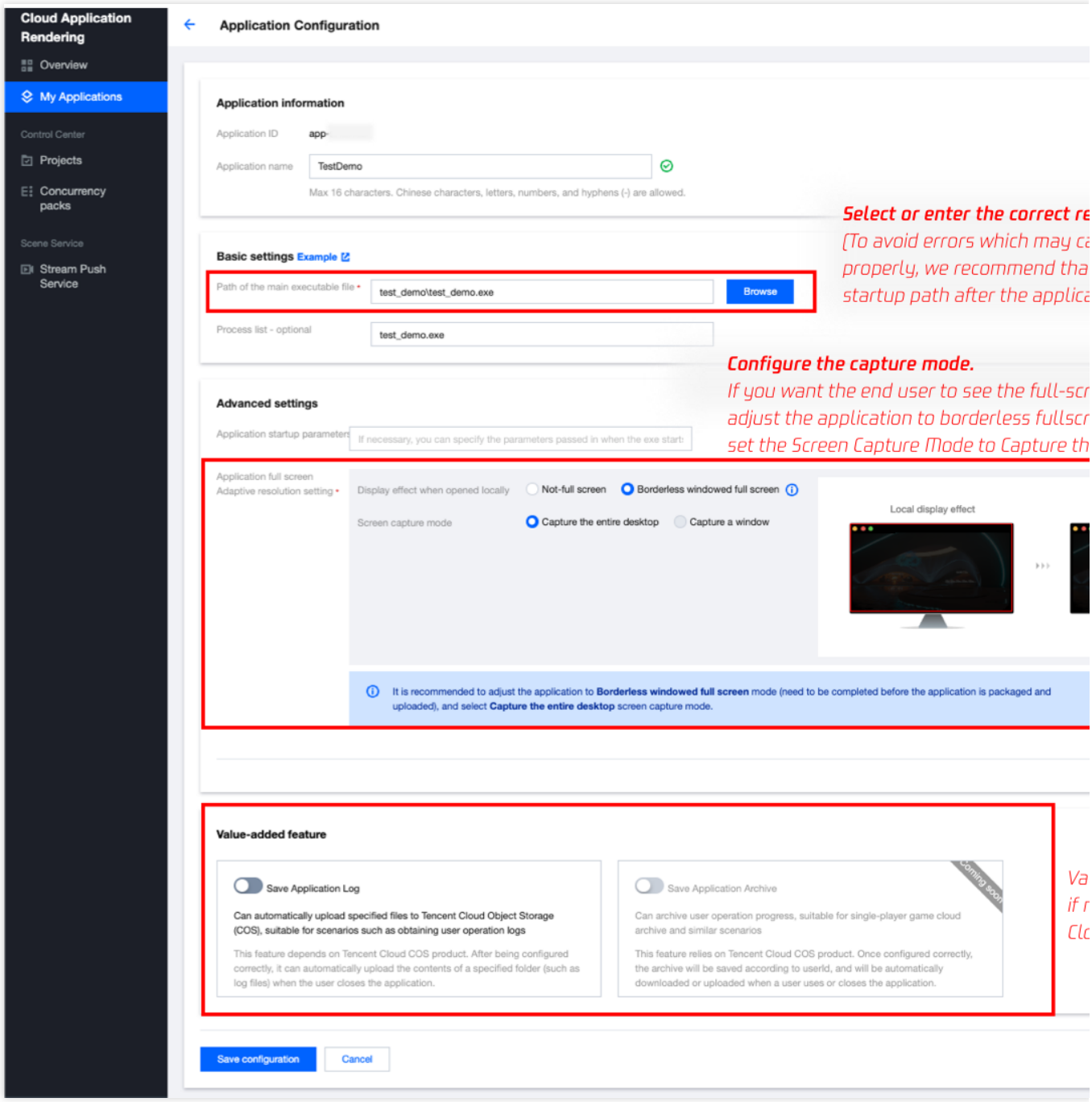
In CAR, you can configure your own startup parameters for your applications. A newly created cloud application can run normally in the CAR environment only after you configure its startup parameters.

Directions

1. Enter the CAR console.
2. Click **My Applications** on the left sidebar, and click **Application Configuration** of the target application.



3. Complete the required information:

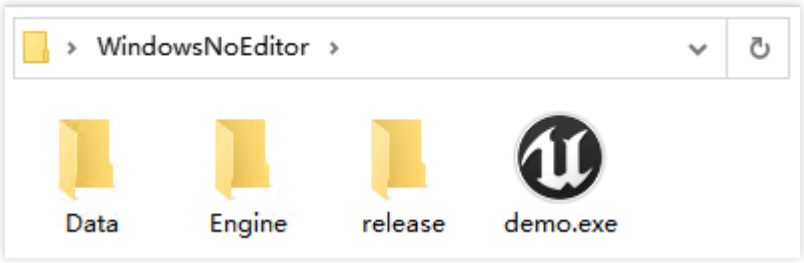


Basic settings

Path of the main executable file

The path of the main executable file of the application is the full relative path of the exe file that launches the application. If your application has been created, you can simply click **Browse** to select and obtain the right path. **Do not select** the path of processes that are not required to start the application, such as `UnityCrashHandler64.exe` .

For example, for the following application package:



The path should be configured as `WindowsNoEditor\\demo.exe`

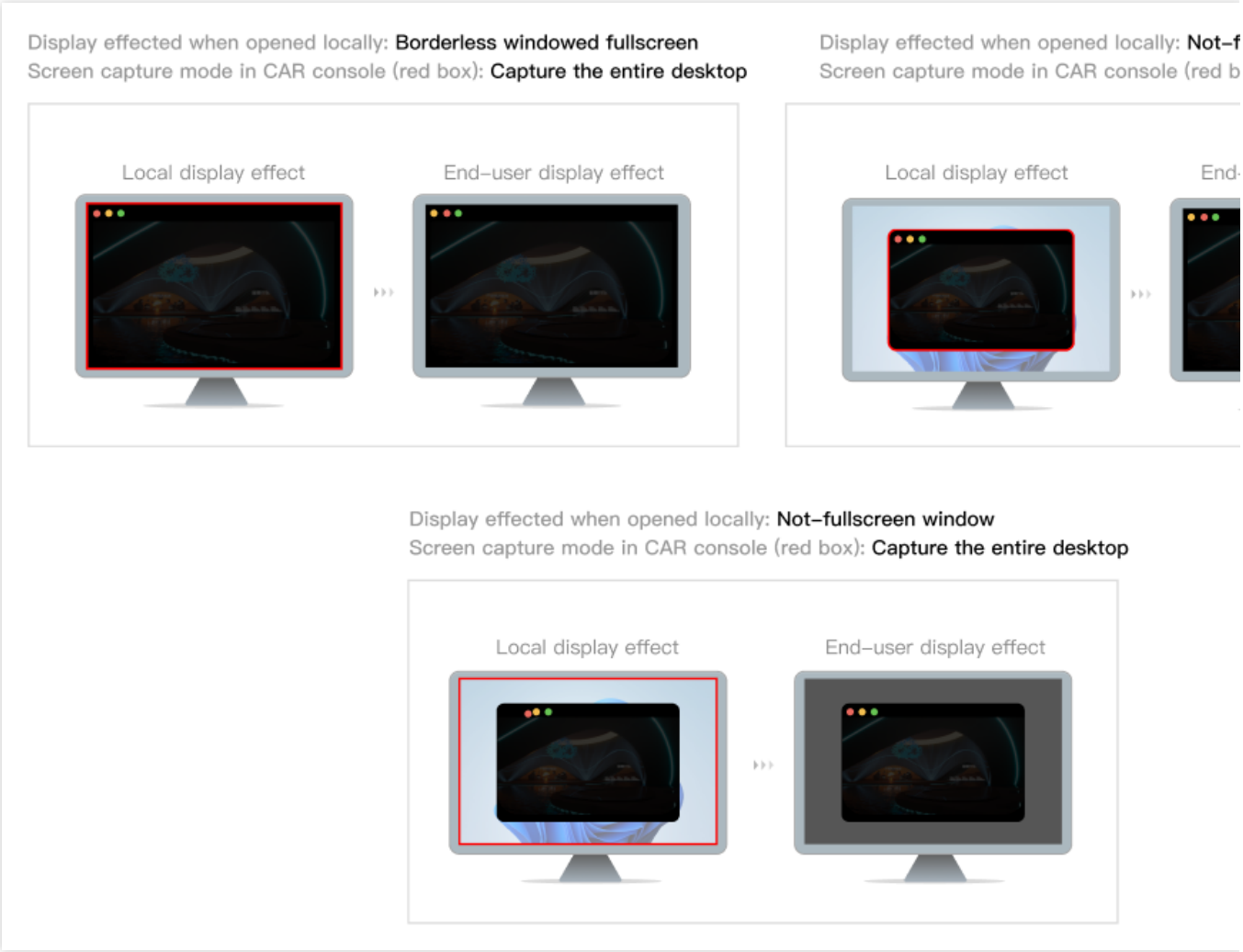
Process list (optional)

We recommend you completely enter all processes to be run by the application, such as `demo.exe|Win64-Shipping.exe` . This way, when a user exits, we will ensure that all filled processes are closed, thereby ensuring stability for the next user who enters. You can run the application on your local PC and open **Windows Task Manager** to check which processes are required.

Advanced settings

Application fullscreen and adaptive resolution setting

CAR provides the adaptive resolution feature which can make the application display full-screen, adapt to the resolution of end-user devices, and have no black borders on the screen.
[Recommend] If you have set the application to a borderless windowed fullscreen mode that can adapt the desktop resolution (have to be done before being uploaded to CAR) according to the User Guides, you can set the `Screen Capture Mode` to `Capture the entire desktop` .
If your application only has a not-fullscreen window mode and cannot be adjusted to borderless fullscreen mode, you can use the `Capture a window` mode. You need to fill in the correct `Window title` and `Class name` according to the instructions. If the window title isn't customized during development, the window title for `Demo.exe` will be `Demo`. If your application is built by Unreal Engine, enter `UnrealWindow` for the Class name.



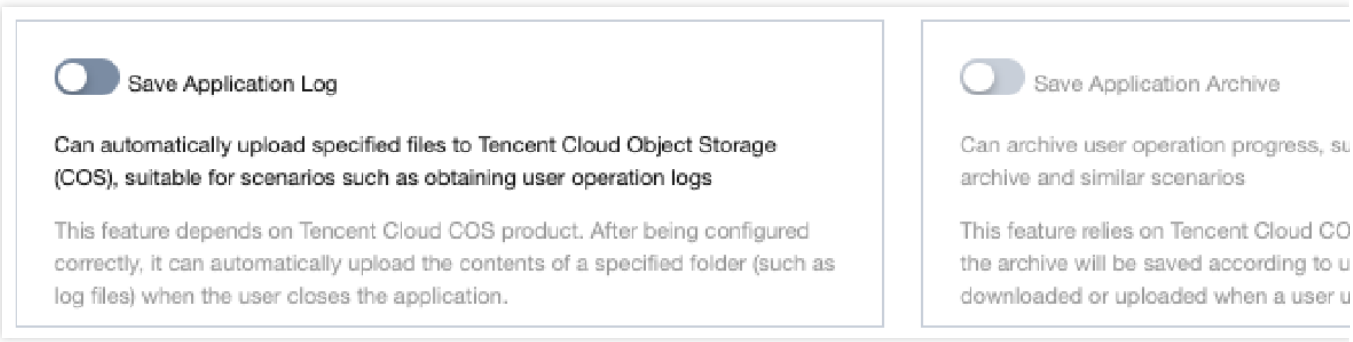
Application startup parameters (optional)

Some applications require command-line (cmd) parameters to be passed in when used locally to achieve effects such as language switching and resolution adjustment. For this, you can fill in the application startup parameters.

Note：

It is recommended to establish communication between the client and cloud applications through a Data Channel, so that the cloud application can obtain the necessary information from the client.

Value-added features (optional)



CAR will automatically clean, reset, and reload the concurrency instance and the cloud application after a user disconnects to ensure data security and stability. Therefore, without configuring value-added features, the user operation logs and usage progress files of cloud applications will not be saved.

If you need to obtain the log files of a cloud application, you can configure the Save Application Log feature

If you need to save the user usage progress of an application, you can configure the Save Application Archive feature

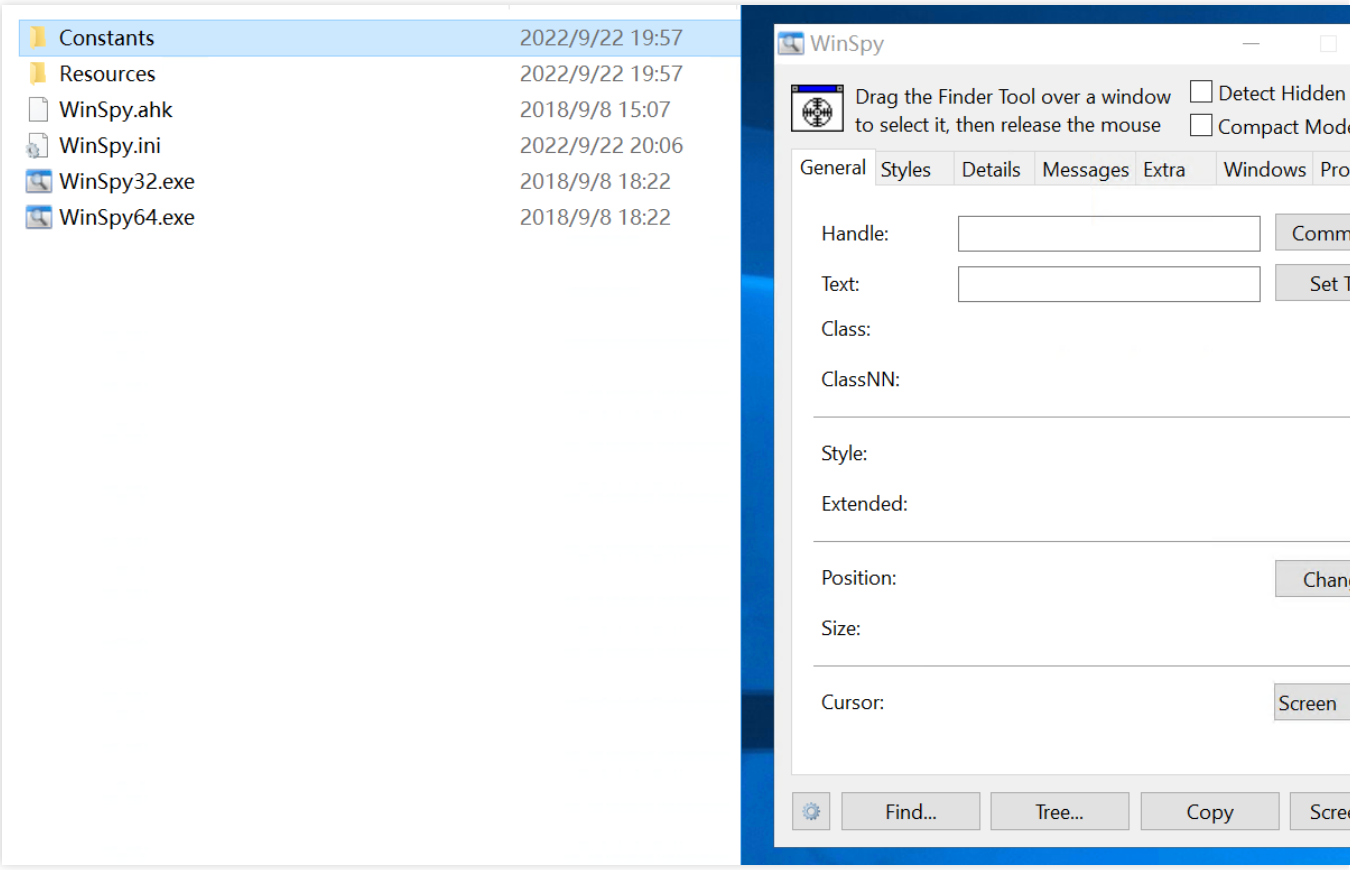
Using Window Capturing Mode

Last updated：2024-01-26 12:00:38

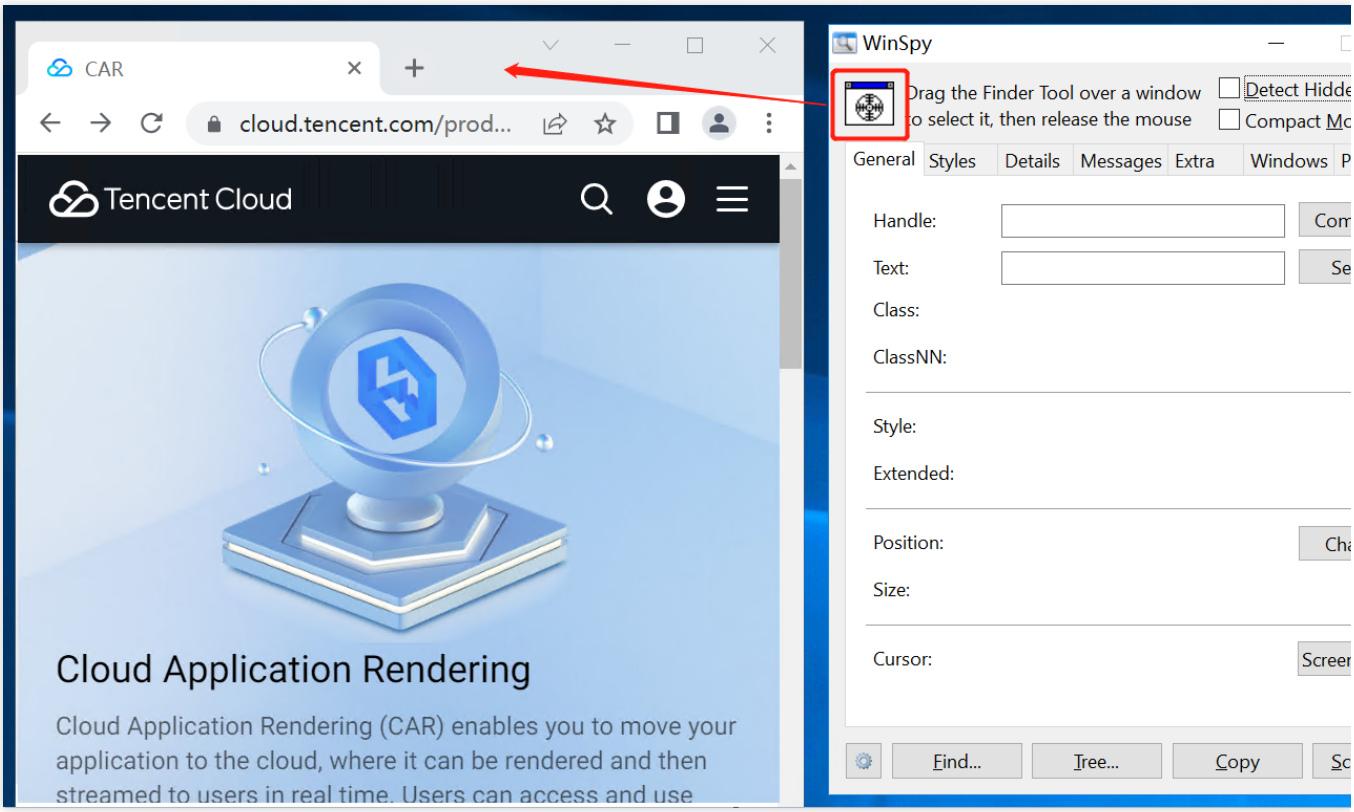
CAR provides a way to capture the window of the cloud application. In this way, the end user will only see the application window and not the rest of the desktop, achieving a full-screen-like effect.
You need to provide the application class name and title (the title is the window title during application startup). If the window title isn't customized during development, the window title after `Demo.exe` starts will be `Demo`. If your application is a UE application, enter `UnrealWindow` for the **Class name**.
This document describes how to use WinSpy to capture the application class name and title for the startup parameter configuration.

Directions

1. Download WinSpy to capture the window attributes.
2. Unzip WinSpy, and open WinSpy32.exe or WinSpy64.exe based on your system type.



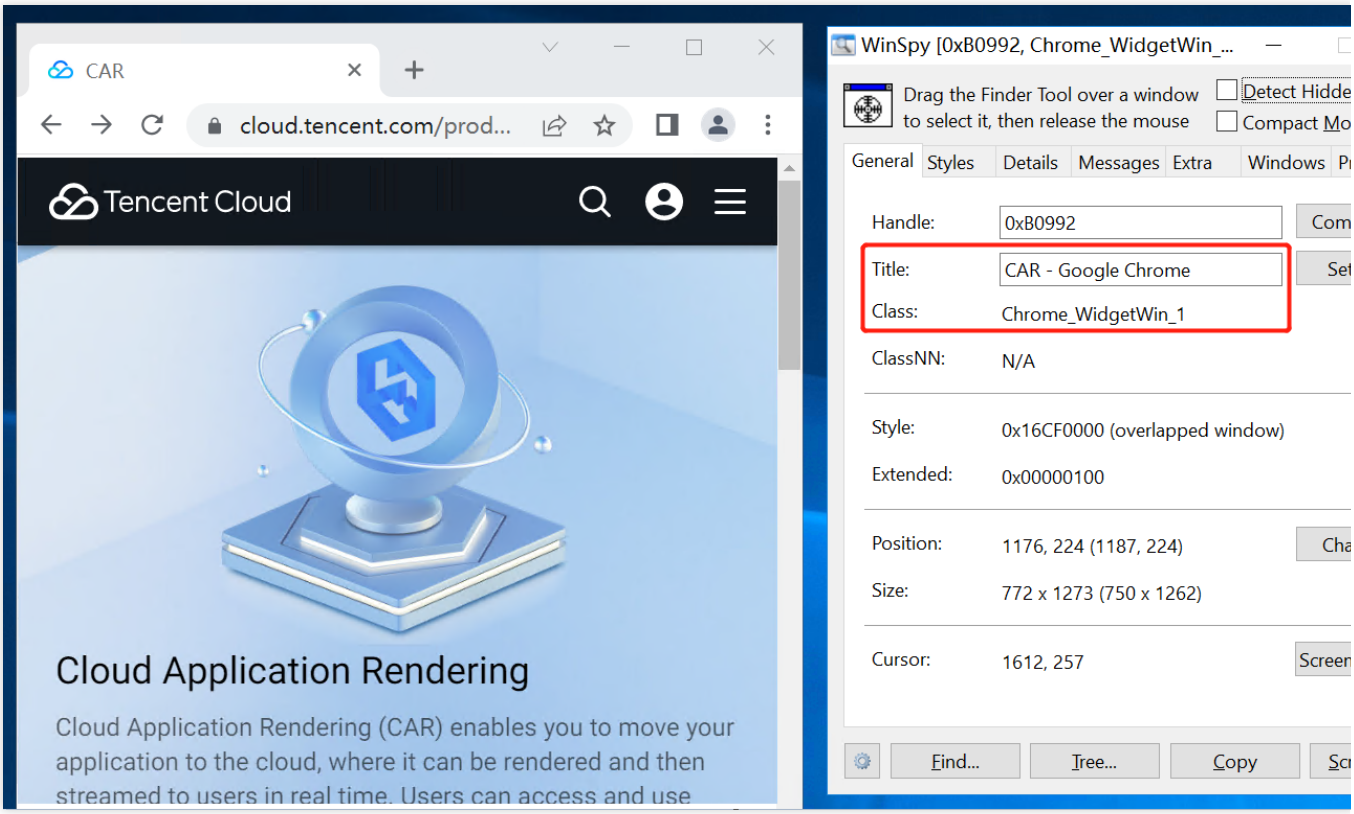
3. Open the application to capture (such as Google Chrome), click and move the Finder Tool (shown in the red box) to the application window, and release it.



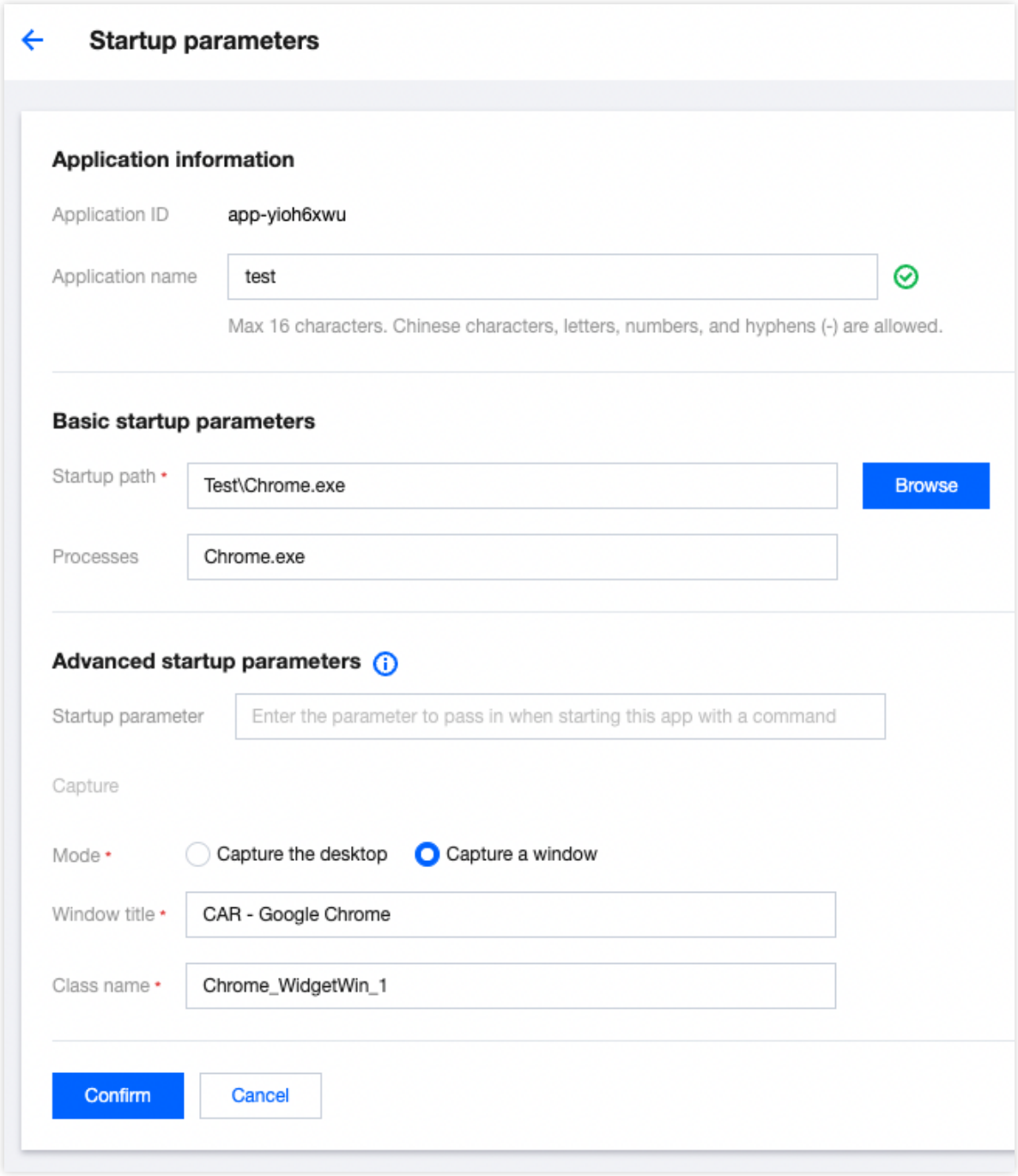
4. WinSpy captures the window attributes.

Title: Window title of your application (such as `CAR - Google Chrome`)

Class: Application class name (such as `Chrome_WidgetWin_1`)



5. Modify the startup parameter configuration according to the captured attributes.



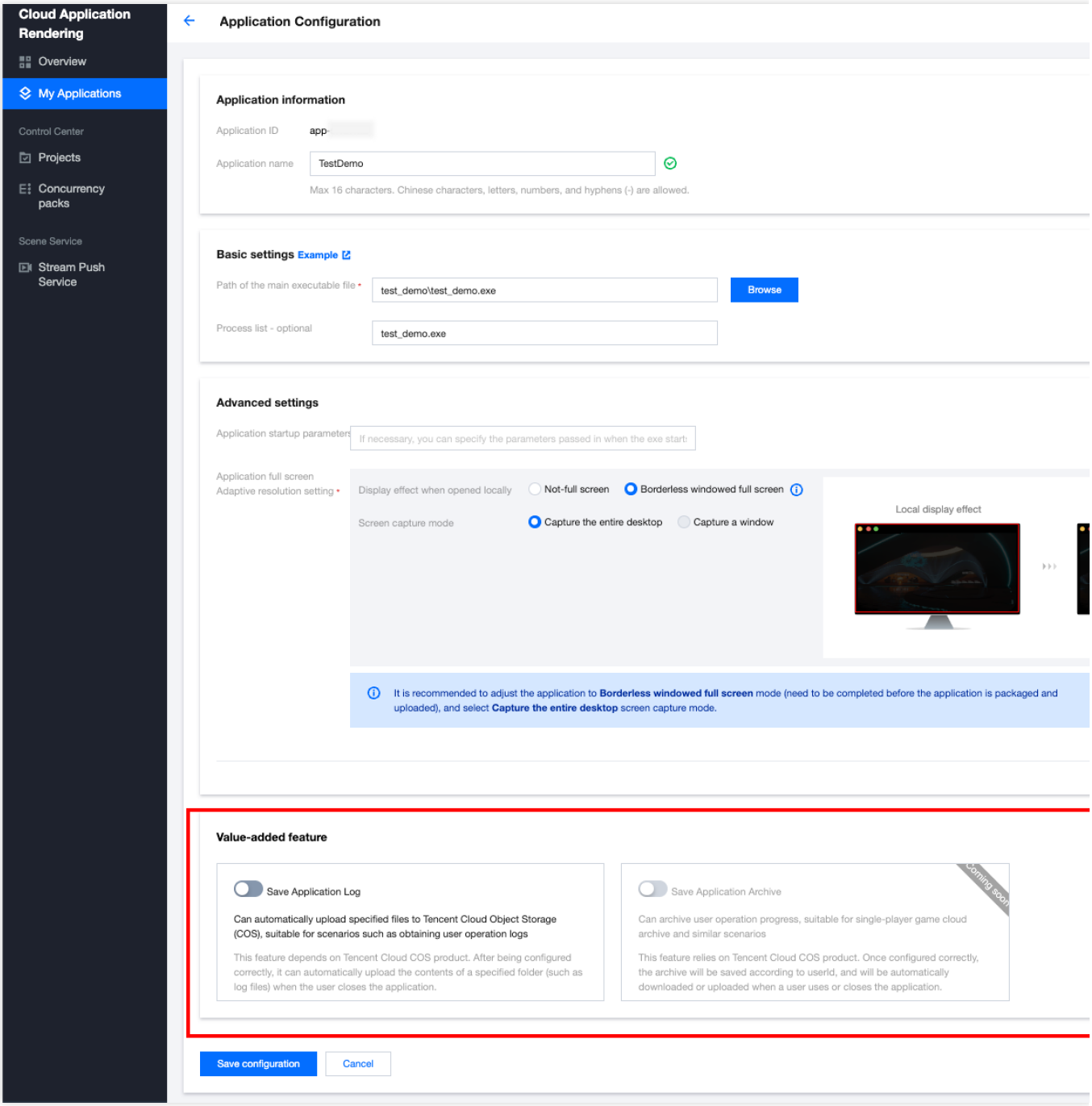
Value-Added Features (Saving Application Logs and Archives)

Last updated：2024-04-17 10:37:45

To ensure data security and application stability, the CAR concurrency automatically cleans, resets, and reloads the application after the user disconnects. This is done to ensure that the next user who connects will not access the data of the previous user. Therefore, if no value-added features are configured, the user's operation logs, usage progress, and other files of the cloud application will not be saved.

To query the log files of a cloud application, you can configure the Application Log Saving feature.

To save the user's application usage so they can continue next time they reconnect, you can configure the Application Archive Saving feature (in beta test).



Note:

Activate COS before using value-added features.

When you use value-added features, possible charges will be incurred by COS instead of CAR. For more information, see COS Billing Overview.

Saving Application Logs

With this feature, a specified folder (such as the Log folder) will be automatically uploaded to the COS bucket after a user disconnects from the application. The basic logic is as follows:

1. In the console, configure the relative path of the folder that needs to be uploaded. Example: WindowsNoEditor\\saved.

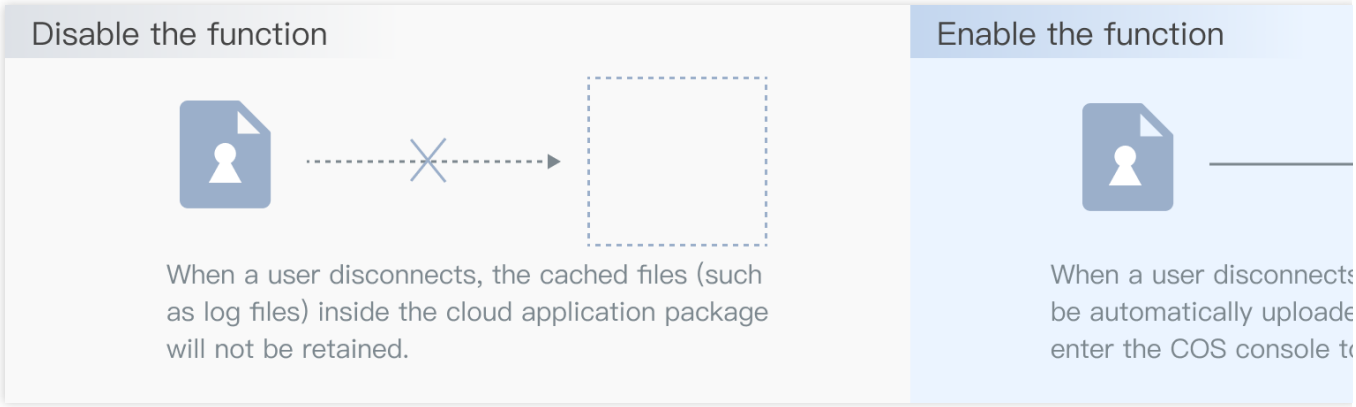
Note:

Only folder paths in the application package are supported. To use a path outside the application package, such as the system disk path, contact Tencent Cloud's architect or your sales rep for assistance.

If you upgrade the application, ensure that the file paths used in the old and new applications are the same; otherwise, the system will be unable to locate the specified file, and the feature will no longer work.

2. When a user connects to the application, it should write log files to the application package directory and saves them locally on the concurrent instance.

3. After the user exits the application, the system will automatically search for the specified file based on the path configured in the console and upload it to the COS bucket.



Saving Application Archives (in Beta Test)

With this feature, a user's application usage will be archived and can be resumed the next time the user connects to the application. The basic logic is as follows:

1. In the console, configure the relative path of the archive folder. Example: WindowsNoEditor\\UserArchive.

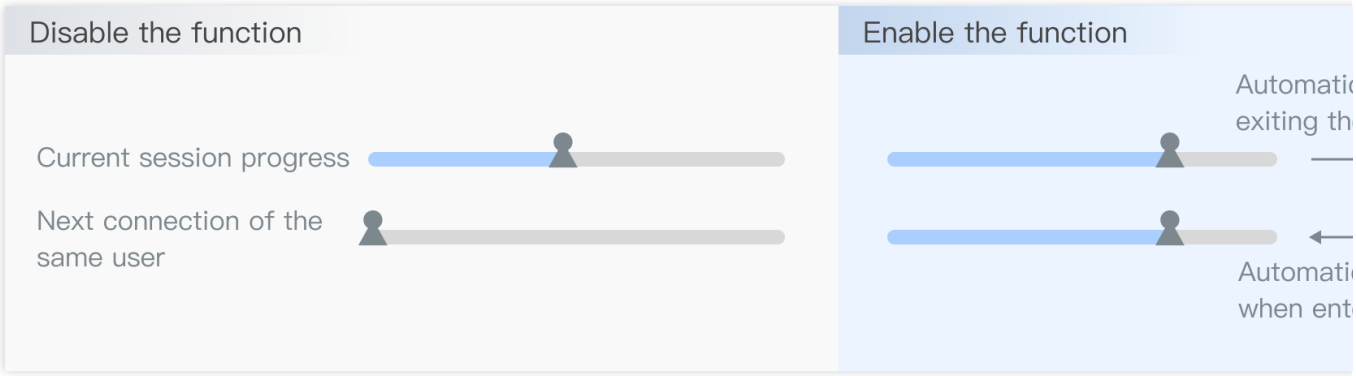
Note:

This feature is in beta test. To try it out, please contact your Tencent Cloud architect or sales rep.

Only folder paths in the application package are supported. To use a path outside the application package, such as the system disk path, contact us for assistance.

If you upgrade the application, ensure that the file paths used in the old and new applications are the same; otherwise, the system will be unable to locate the specified file, and the feature will no longer work.

2. When a user connects to the application, the system will automatically read previous archives from the COS bucket based on the `UserId` .
3. If the user has any archives existing, then before the application starts, the archive folder will be automatically downloaded to the relative path you configured in the console.
4. The data of all the operations the user performs on the application will be archived locally on the concurrent instance.
5. After the user exits the application, the system will automatically upload the archived data to the COS bucket.



How to View the Saved Logs or Archives?

Assume that you have an application with the ApplicationId "app-1234abcd"
The user's UserId is "user123"
The RequestId of the last session is "01fdc815-c4e7-4642-819e-a011856dfd5a1"
The timestamp of the last session is "1709284736"
Then, **the storage path in COS should be:**
Save Application Log: AppLogs/app-1234abcd/01fdc815-c4e7-4642-819e-a011856dfd5a1.zip
Save Application Archive **(in beta test)**: userData/app-1234abcd/user123-1709284736-tx

FAQs

What are UserId and RequestId?

`UserId` is the custom unique user identifier passed in to CAR, such as `user123456` . A `RequestId` , such as `01fdc815-c4e7-4642-819e-a011856dfd5a1` , is returned when CAR requests a TencentCloud API.

How do I view the RequestId?

If you are using a test or quick launch link, you can open the debug panel in the toolbar displayed on the application's main page (Ctrl+～) and view the `RequestId` information at the bottom.
If you use Chrome DevTools, you can get the `RequestId` of `CreateSession` on the **Network** tab.
If you have integrated to the CAR SDK, the returned values of the TencentCloud API include `RequestId`. We recommend you record it on the business backend.

Application Update and Version Management

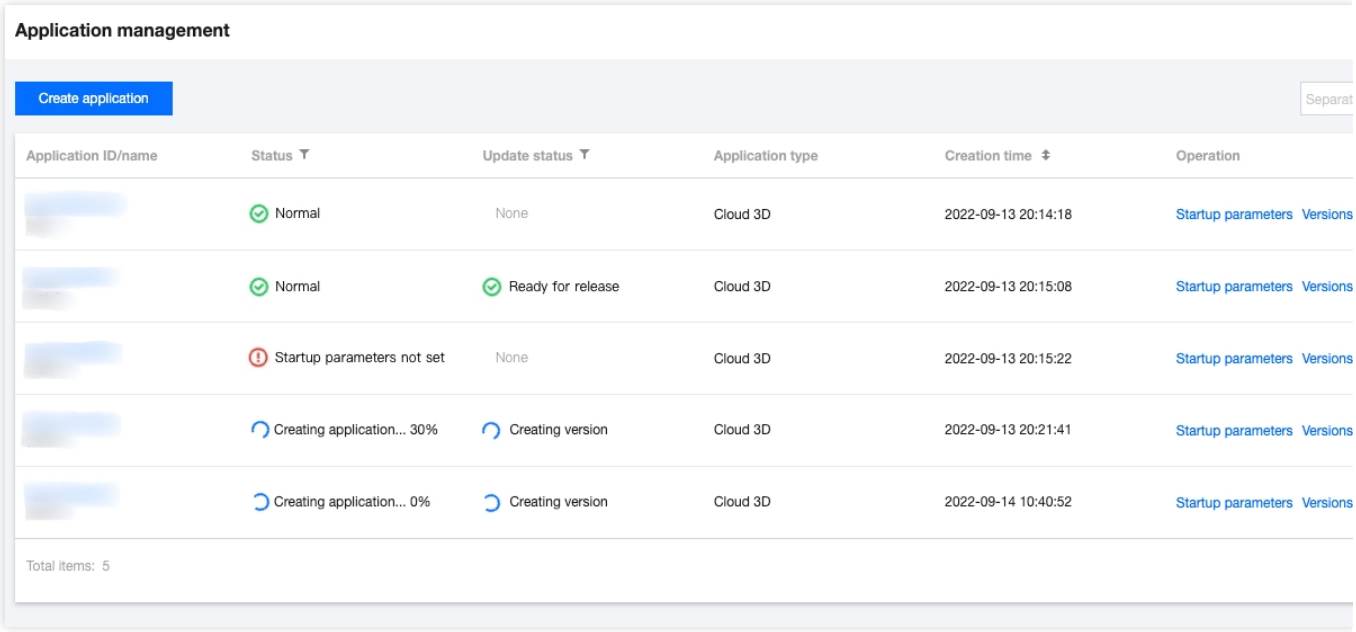
Last updated：2024-01-26 12:00:38

CAR offers version management capabilities. Under the same application ID, you can maintain up to **five** versions and perform operations such as new version upload and release as well as version rollback.

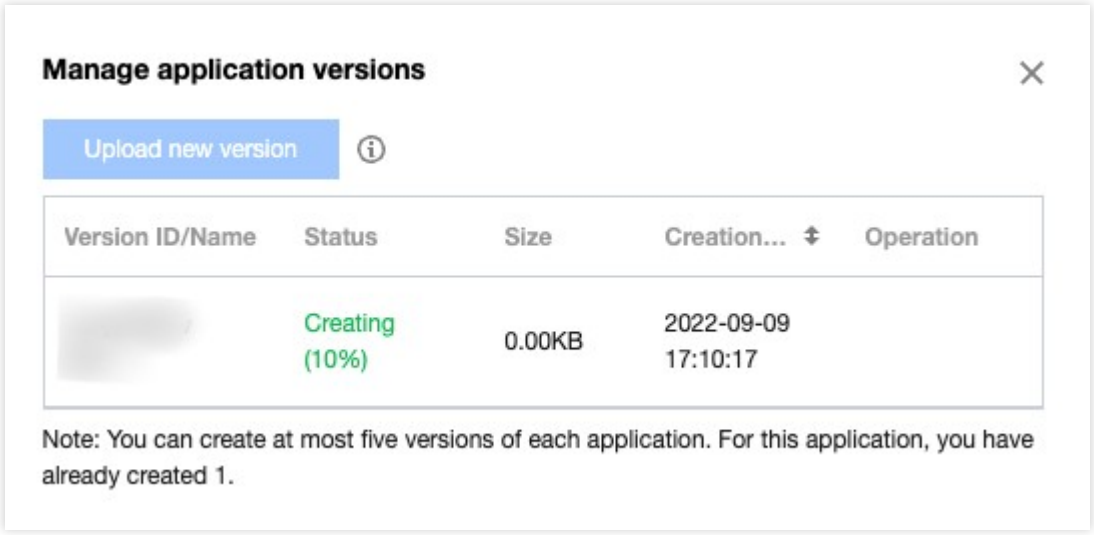
Application version Update

Note:
During application update, make sure that the application startup path is the same as that of the original application package; otherwise, you need to modify the startup parameter configuration. The URL must be a URL that can be accessed and trigger download directly. Download URLs of cloud disks or other URLs requiring verification are not supported.

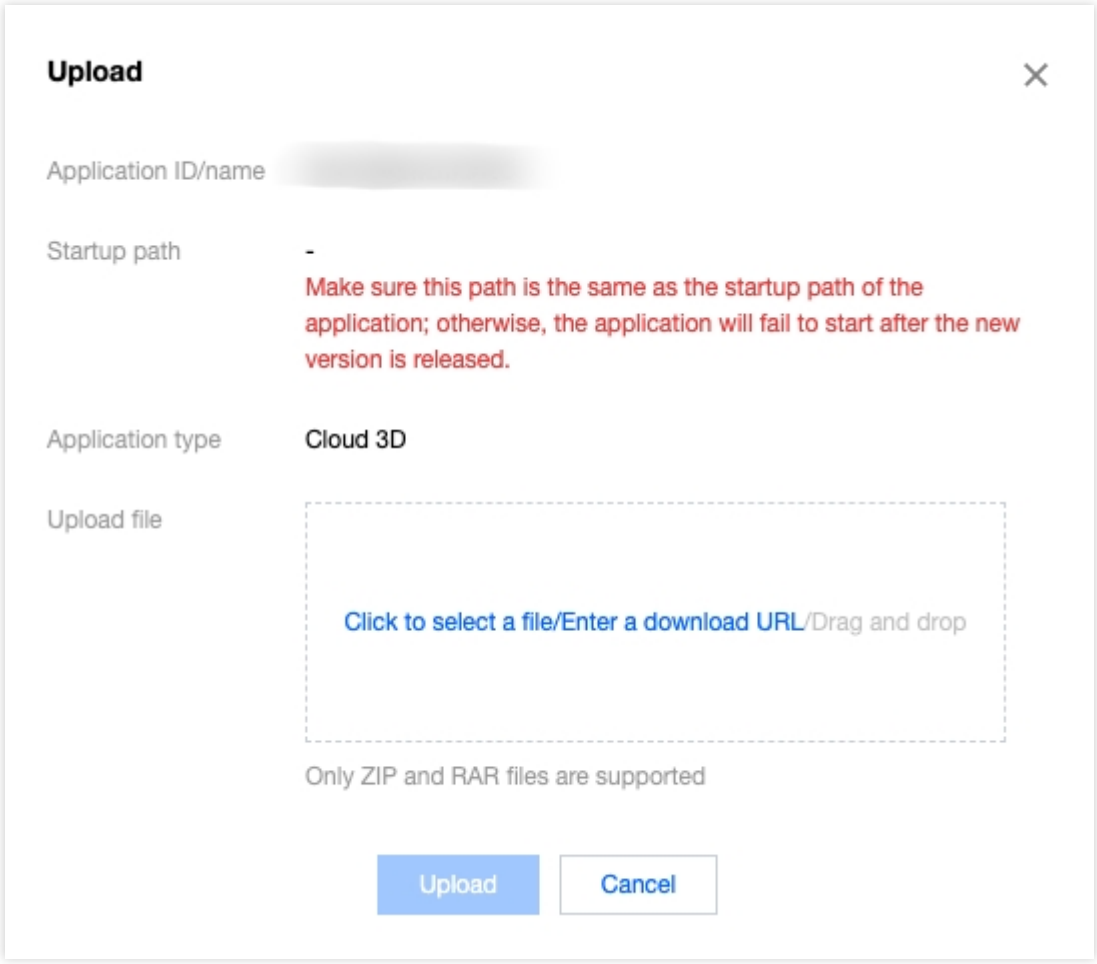
1. Go to the CAR console.
2. Click **Application management** on the left sidebar and click **Versions** on the **Application management** page.



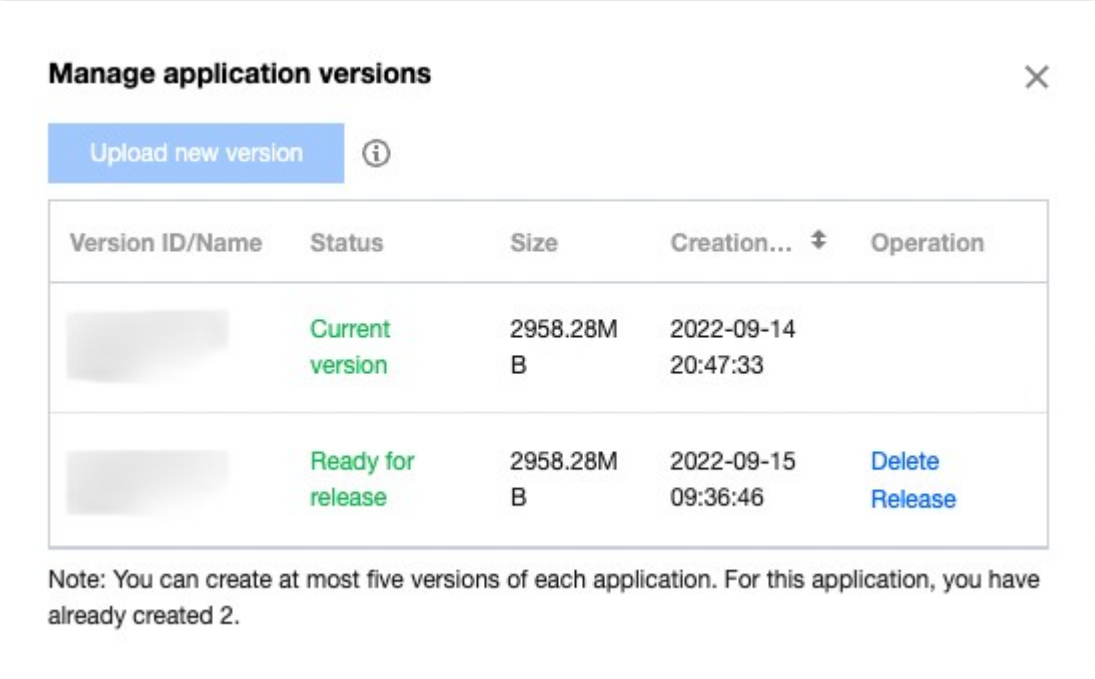
3. In the **Manage application versions** pop-up window, click **Upload new version**.



4. Select an upload method. You can select a file, drag a file, or enter a URL to upload a file.



5. After submission, the application version will be created immediately. You can view the upload progress in **Manage application versions**.



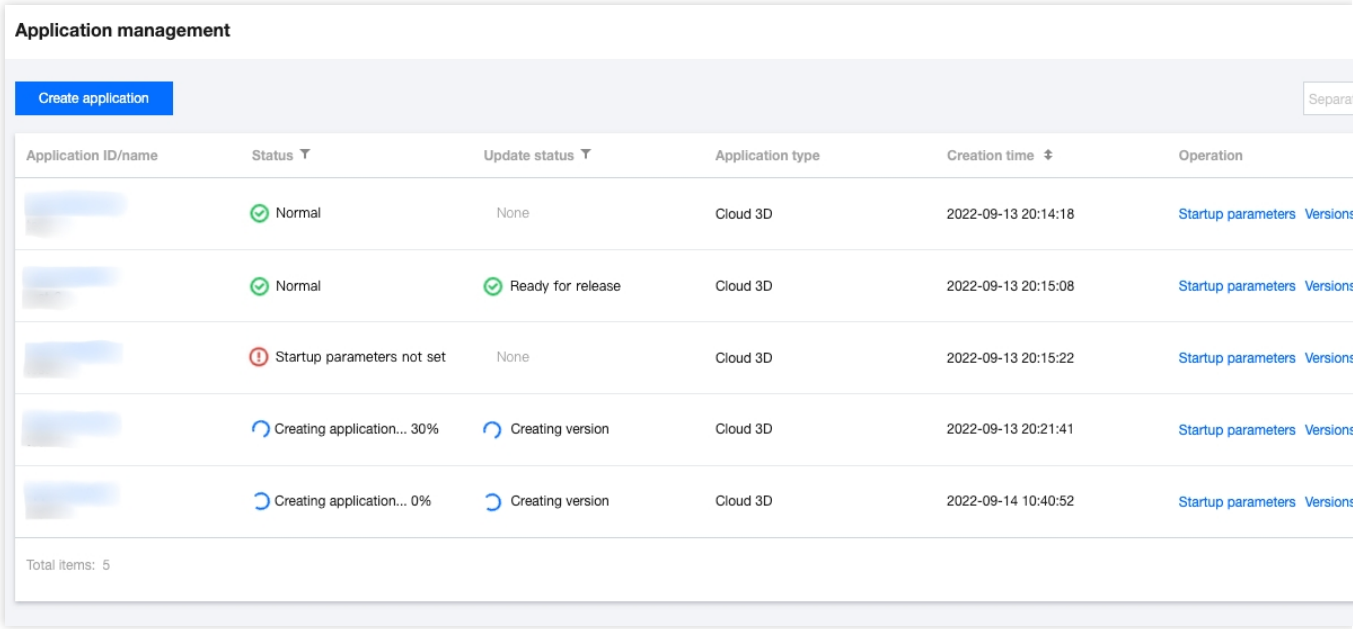
6. After the creation, you can test the new version in the business environment based on the version ID. After the test is completed, click **Release**.

Application version Rollback

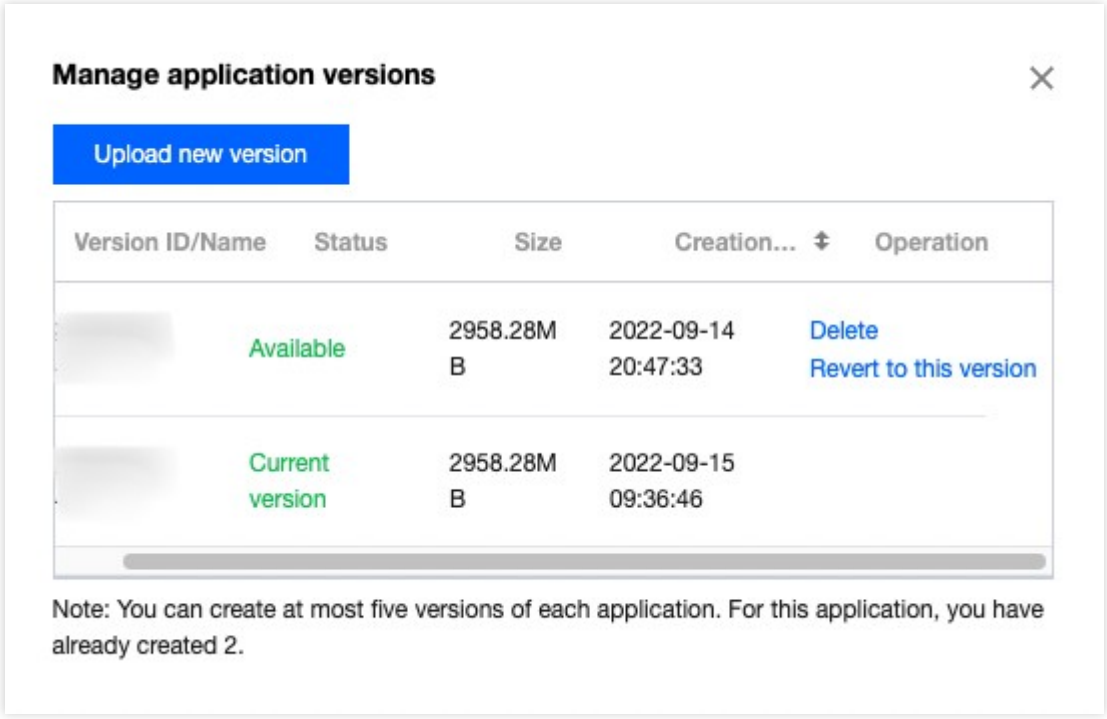
Note:

During version rollback, make sure that the startup paths of different versions are the same; otherwise, the application will be unavailable during version switch as the startup parameter configuration doesn't match the version startup path.

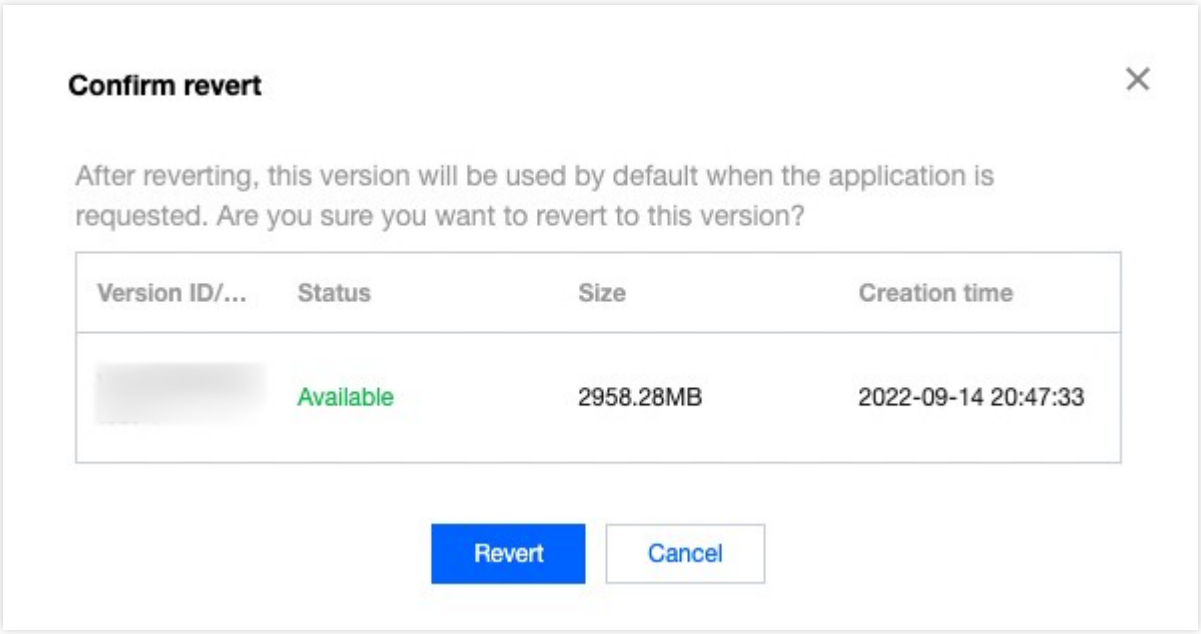
1. Go to the CAR console.
2. Click **Application management** on the left sidebar and click **Versions** on the **Application management** page.



3. In the **Manage application versions** pop-up window, select an earlier version and click **Revert to this version**.



4. In the **Confirm revert** pop-up window, confirm that everything is correct and click **Revert**.



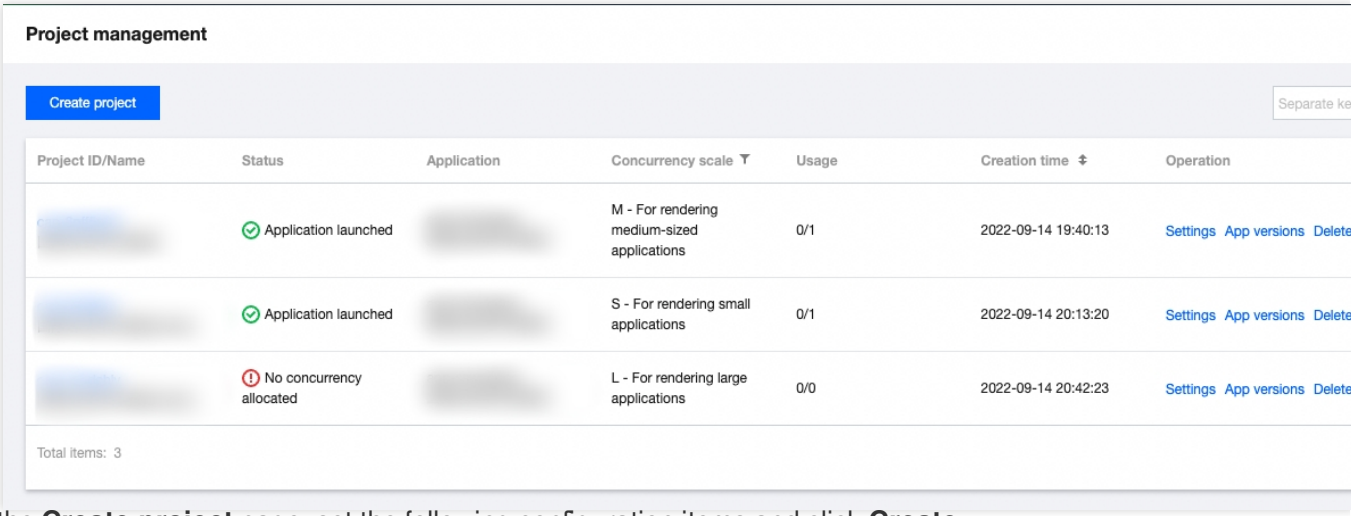
Project Management
Creating a Project

Last updated：2024-01-26 12:00:38

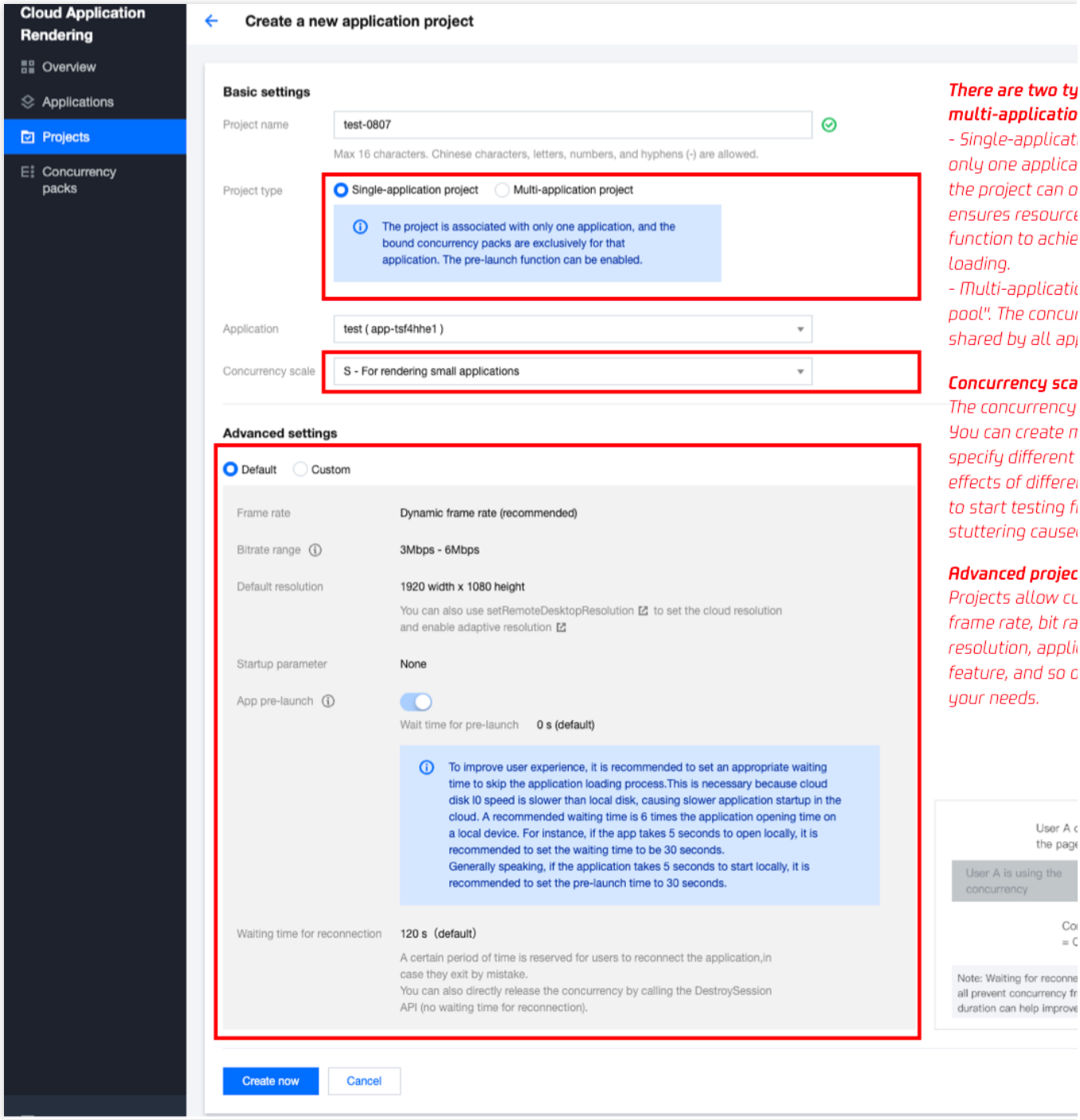
In CAR, concurrencies are scheduled for applications based on which project the application is associated with. After you upload an application, you need to associate it with a project. Then the application can use the concurrency packs (computing resources) that are bound to that project.

Directions

1. In the CAR console, click **Projects** on the left sidebar and click **Create project**.



2. On the **Create project** page, set the following configuration items and click **Create**.



Basic settings

Configuration Items		Description
Basic settings	Project name	A custom project name. It can contain up to 16 letters, digits, or hyphens (-).
	Project type	Single-application: The project is associated with only one application, and all concurrency packs bound to the project are dedicated to that one application. The prelaunch feature can be enabled.

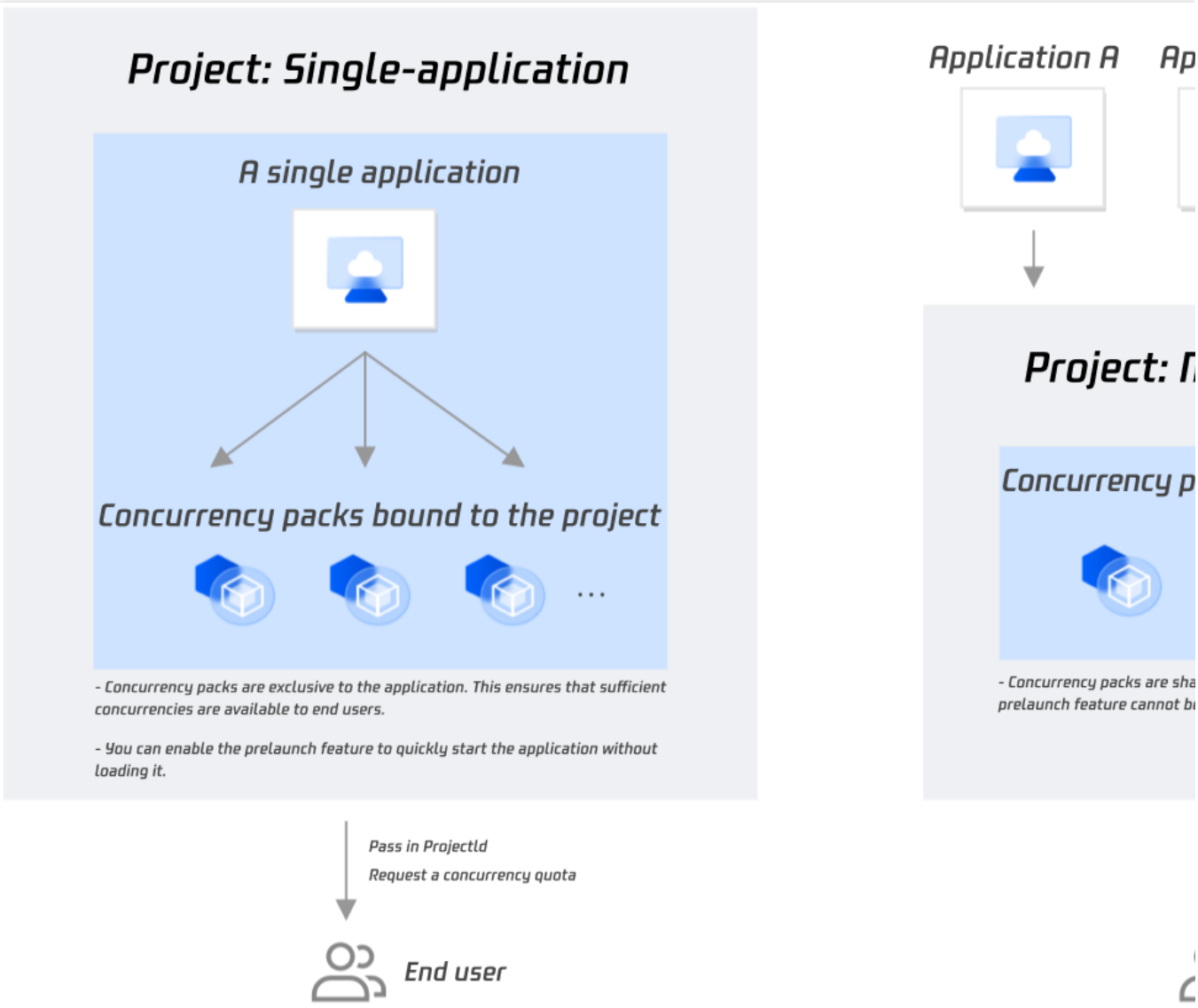
		Multi-application: Concurrency packs that are bound to the project are shared by all applications within the project, but the prelaunch feature cannot be enabled. For more information, see Project types.
	Application	Select an application to be associated with the project.
	Concurrency Scale	Select the concurrency scale for the project.

Project types

CAR supports two types of projects:

Single-application project: The project is associated with only one application, and all concurrency packs bound to the project are dedicated to that one application. This guarantees the availability of resources, and you can enable the prelaunch feature to quickly load the application when a user connects to it.

Multi-application project: Like a resource pool, concurrency packs bound to the project can be shared by all applications. When an end user makes a request, idle concurrencies can be scheduled from the pool in real time. For more information, see How to Implement Concurrency Sharing Through Multi-Application Project.



Project Type	Description	Main Strengths	Use Case
Single-application	The project is associated with only one application, and concurrency packs bound to the project can be used only by that application. When you call the ApplyConcurrent API to request a concurrency, you need to pass in the `ProjectId` parameter.	You can enable the prelaunch feature to quickly load the application when a user connects to it.	A virtual exhibition application requires at least 100 concurrencies and uses the prelaunch feature to allow users to quickly access the application without waiting for it to load.
Multi-application	Concurrency packs that are bound to the project are shared by all the applications under the project, but the prelaunch feature cannot be enabled.	Multiple applications can share concurrencies	Multiple exhibition applications which are independent of each other need to share the concurrency pool, so that each application can get idle

	When you call the ApplyConcurrent API to request a concurrency, you need to pass in **both the `ProjectId` and `ApplicationId` parameters**.	to reduce costs.	concurrencies from the pool as needed when there are user access requests.

Note:

To maintain a balance between user experience and cost optimization, you can create a single-application project and a multi-application project and use them together:

When there are user requests, you can first schedule the concurrencies in the single-application project. If prelaunch is enabled for the project, users can quickly use the application without waiting for it to load.

When available concurrencies in the single-application project become insufficient (ApplyConcurrent() returns a prompt indicating that **there are no idle concurrencies**), you can then schedule concurrencies in the multi-application project.

For more information, see Starting an Application and Queue Feature.

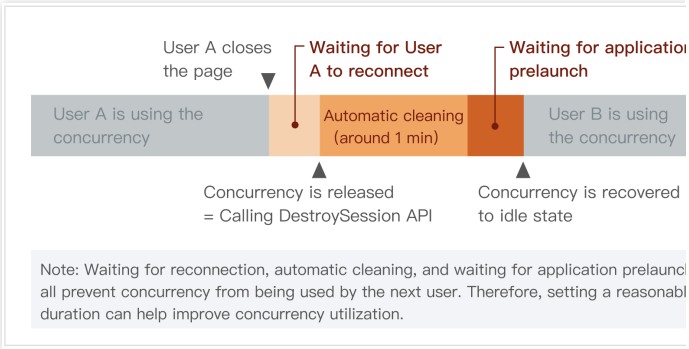
Advanced settings

Projects allow customization of parameters such as frame rate, bitrate range, default cloud desktop resolution, application startup parameters, pre-launch, wait time for reconnection, etc. You can adjust these parameters according to your needs.

You can create different projects to set different parameters for the same application. For example, you can create two projects for the same application, one project with S-type concurrency, 30FPS, and 3-6Mbps bitrate range, and the other project with L-type concurrency, 60FPS, and 5-8Mbps bitrate range. This way, you can provide the first project to ordinary users and the second project to VIP users.

Configuration Items		Description
Advanced settings	Frame rate	Dynamic frame rate (recommended): adjusts the actual encoding frame rate based actual rendering frame rate of the cloud application. Specified frame rate: supports options of 30FPS and 60FPS. If you have other requi please contact your Tencent Cloud sales rep for assistance.
	Bitrate range	The bitrate range is related to the concurrency scale, and the default values are as f S - For rendering small applications: 3-6Mbps M - For rendering medium-sized applications: 4-8Mbps L - For rendering large applications: 5-8Mbps XL - For rendering extra large applications: 7-10Mbps It is not supported to exceed the maximum bit rate value specified by the concurrenc you have special requirements, please contact your Tencent Cloud sales rep for ass
	Default resolution	That is the default desktop resolution of the cloud concurrency instance, which can b to 3000 x 3000. If you have higher resolution requirements, please contact your Ten

	Cloud sales rep for assistance. If you need to adapt to the resolution of different term devices, please refer to the adaptive resolution documentation.
Startup parameter	The startup parameters configured in the project will override the startup parameters configured in the application. If not filled in, the startup parameters filled in the applic configuration will be used by default.
App prelaunch	This feature allows you to open the application in advance in the cloud concurrency (without any additional fees) and complete the loading process before the user uses achieving instant application startup. This feature is only available for single-applica applications and is not supported for multi-application type projects.
Wait time for prelaunch	Cloud disks have slower IO speed compared to local disks, causing slower startup of applications in concurrencies. To ensure a smoother user experience, it's recommended to enable the prelaunch function and set an appropriate wait time to allow prelaunch to complete before connecting the next user. For instance, if the app takes 5 seconds to open locally, it is recommended to set the waiting time to 30 seconds.

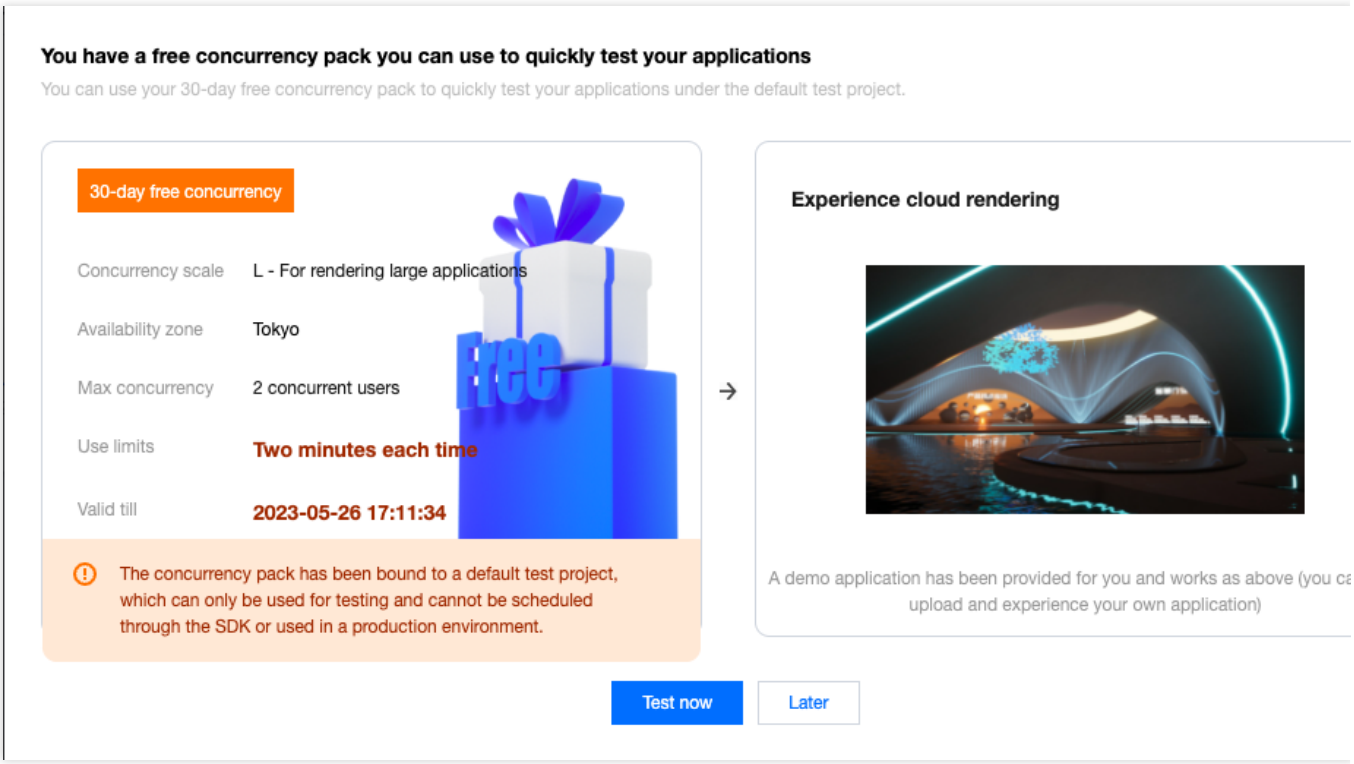


	Waiting time for reconnection	To prevent users from accidentally exiting the application, a certain duration is reserved for users to re-enter the application. You can also maintain user connection status based on heartbeats in the backend, and release the concurrency directly by calling the DestroySession API (without waiting for reconnection).	

Default Test Project and Free Concurrency Pack for New Users

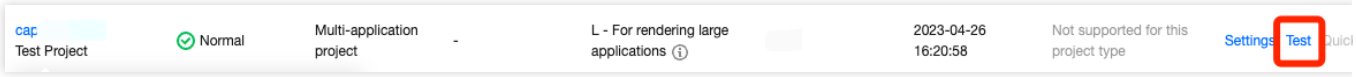
Last updated：2024-01-26 12:00:38

CAR offers new users a 30-day free concurrency pack of two L concurrencies.
The free concurrency pack offered to new users is bound to the default test project, which you can use to **test the uploaded application** for free.
Note:
Please choose an availability zone closer to the testers to avoid lagging problems caused by the distance. For instance, If you are in Japan or Seoul, you will need to select the Tokyo region. The 30-day free concurrency pack only support a few available zones for now, please refer to Billing for all availability zones supported by CAR.
The default test project and the free concurrency pack offered to new users are intended for testing only. **They can be used by each user for two minutes each time and cannot be scheduled through the SDK or used in a production environment.**

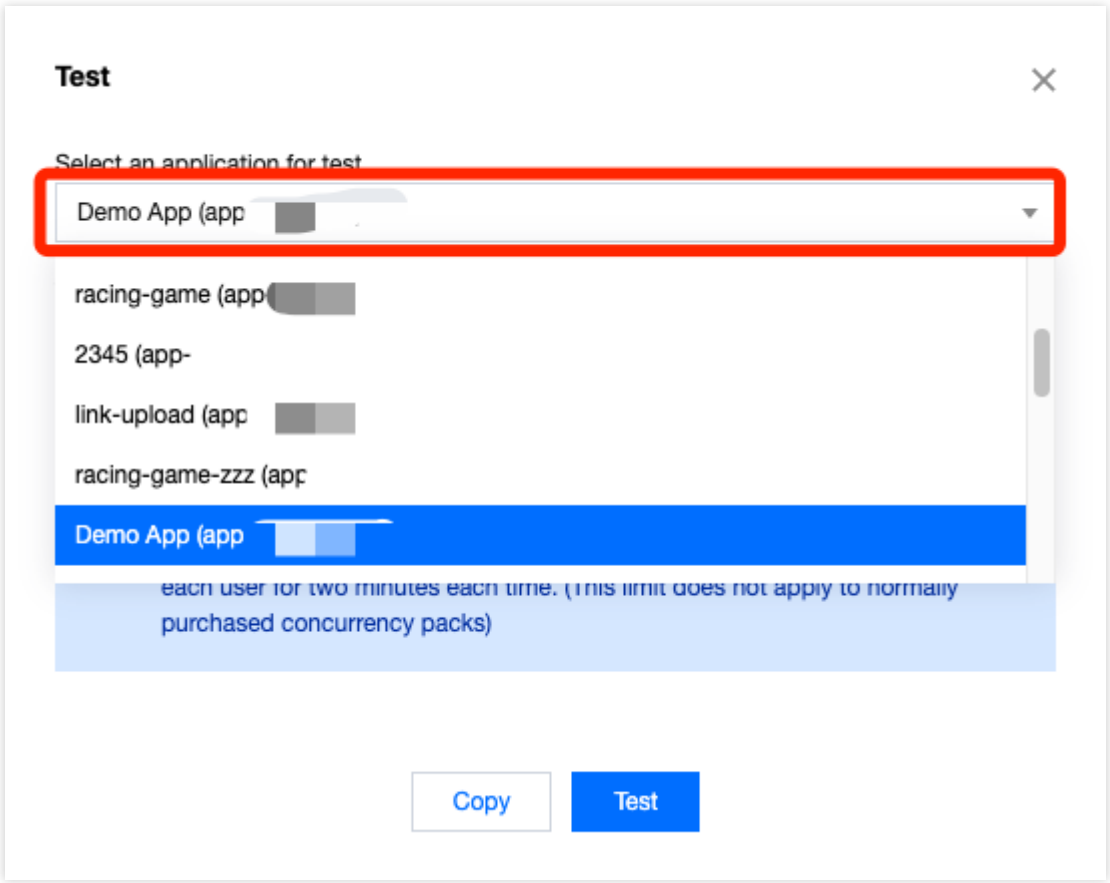


Directions

1. Click **Test now** in the pop-up window or go to the **Projects** page, find the default test project, and click **Test** under the **Operation** column.



2. Select the demo application or your uploaded application to generate a test link and password. If you select your uploaded application, you will first need to correctly configure the application startup parameters.



Note:

If you need to test multiple applications, you can click **Test** multiple times and select a different application each time to generate a new password. Make sure to save the generated information locally.

A test password is valid for seven days. Each time you click **Test**, a new password will be generated, but the previously generated passwords are still valid if they haven't expired.

While a test password is still valid, you don't need to generate a new one to perform operations such as purchasing a concurrency pack and updating the application version.

3. Start a test as instructed in Test.

Effect of the demo application

Common issues encountered during testing:

A message is displayed indicating that there are no idle concurrencies:

Check whether there are any idle concurrencies under the project in the console. When a user exits the application, **it takes about one minute for the concurrency to be automatically cleared. Only then will the concurrency become idle so that a new user can connect to it**.

The application cannot start:

We recommend that you first check whether the application startup parameters are correctly configured.

If the application uses window capturing mode, check whether the window title and class name are correctly entered. If they are configured incorrectly, a black screen will occur.

The application starts slowly:

Generally, the prelaunch feature is used to load an application in advance so the application will already be running when a user connects to it. Because multi-application projects do not support prelaunch, the applications under those projects take a longer time to start.

For a single-application project, you can enable prelaunch so that the application will be loaded in seconds when the user connects to it.

When a user exits the application, the concurrency they were connected to will be repossessed, cleared, and reset, and when the concurrency becomes idle, the application will be prelaunched again. If the next user enters the

application just when the concurrency becomes idle, the application may have not been completely prelaunched.

The delay data is abnormal: We recommend that you first check for local network jitter. You can try accessing over a 4G/5G network.

The application is slow: This may be because your application has high requirements for computing power but the concurrency scale is low (such as S). We recommend you try a higher concurrency scale.

If your problem still persists, contact us for assistance. For more FAQs, see Cloud Application FAQs.

Test

Last updated：2024-01-26 12:00:38

The test feature lets you experience the basic effects of operating your application in the cloud and helps you select the most appropriate concurrency specification for your application. To meet your special business requirements and guarantee the optimal user experience, you also need to set up your own client program and backend service and connect to CAR's backend APIs and SDKs. You can use the demo to quickly launch your business.

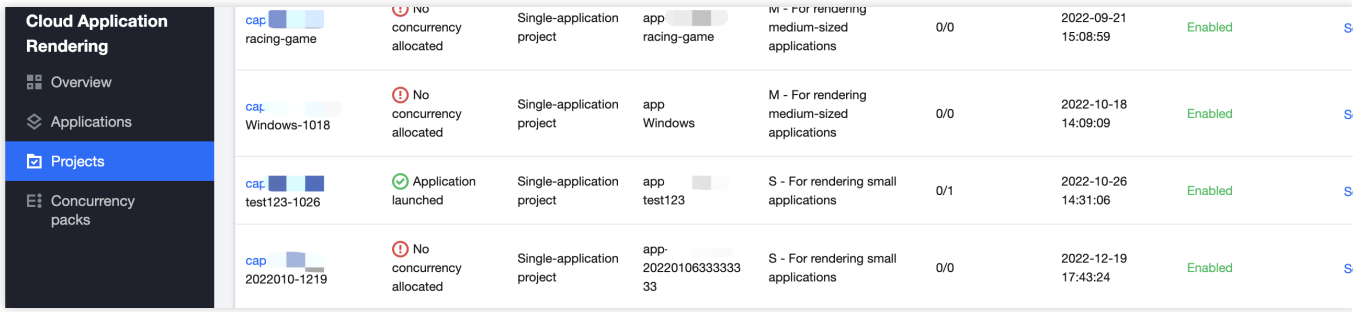
Directions

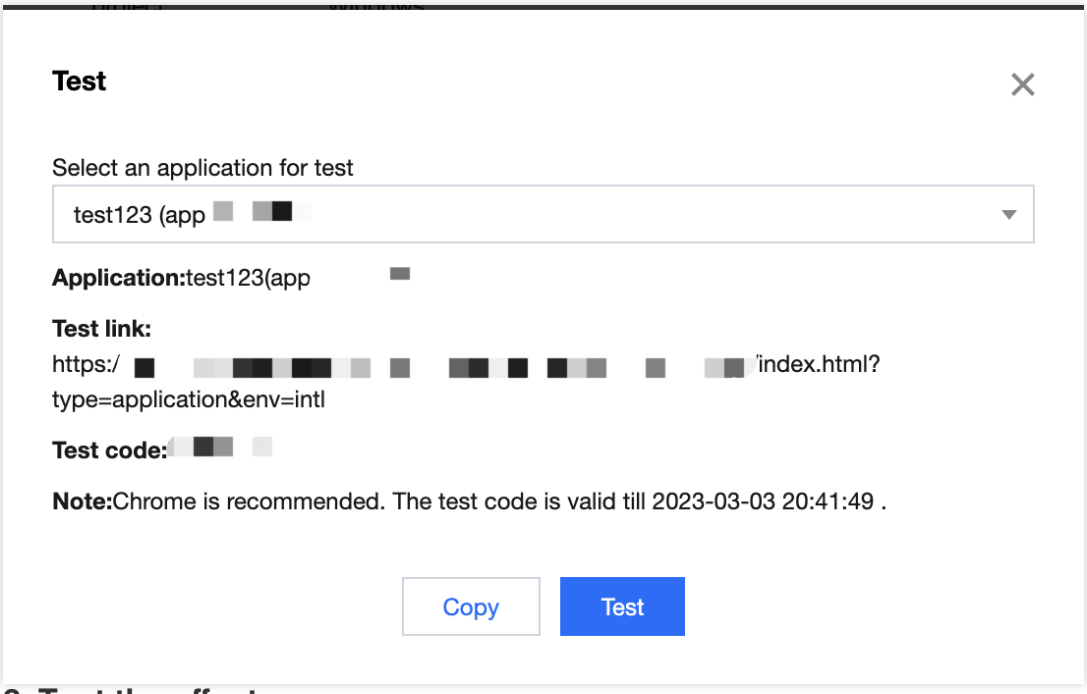
Prerequisites

You must perform the following steps before you can start a test:
1. Upload an application and correctly configure the application startup parameters.
2. Create a project, associate it with the uploaded application, and specify the concurrency specification.
3. **Purchase a concurrency pack suitable for the project** and bind it to the project.
After completing the above steps, you can go to the **Projects** page n the CAR console to test the application.

Step 1. Click Test to generate a test link and password

After you click **Test**, a test link and password will be generated automatically.
Note:
A password is valid for seven days. Each time you click **Test**, a new password will be generated, but the previously generated passwords will still be valid if they haven't expired.
While a password is still valid, you don't need to generate a new one to perform operations such as purchasing a concurrency pack and updating the application version.





Step 2. Test the effect

Open the test link and enter the password to access the cloud application. This lets you experience the effect of accessing and operating the application online. You can test different concurrency specifications to find the most appropriate one for your application.

Open the debug panel (Ctrl+～) in the toolbar on the Test page. You should pay attention to the following:

FPS (frame rate): In normal cases, the frame rate should be kept above 30. If you find that the frame rate suddenly drops when you enter certain scenes or perform certain operations, it may be due to a sudden increase in GPU computing power consumption that exceeds the capacity of the current concurrency specification and results in frame drop. In this case, we recommend you try using a higher concurrency specification.

RTT (round-trip time): When the RTT value exceeds 100 ms, it may result in noticeable latency. We recommend you first check for local network jitter. You can try accessing over a 4G/5G network. Additionally, if the concurrency region is too far from your physical location, it can also result in high RTT.

Region: The region where the concurrency is located. CAR can automatically schedule the nearest available concurrency based on the IP address of the end user. If you find that the concurrency region is too far away, it may be due to insufficient number of available concurrencies that are closer to the end user. You can contact us for assistance.

InstanceType (concurrency specification): CAR supports four concurrency specifications: S, M, L, and XL, which are suitable for rendering small, medium-sized, large, and extra large applications. For details, see the official billing documentation. If your application requires high computing power but is using a lower concurrency specification, it may lead to high CPU/GPU usage, causing lags or forced quits.

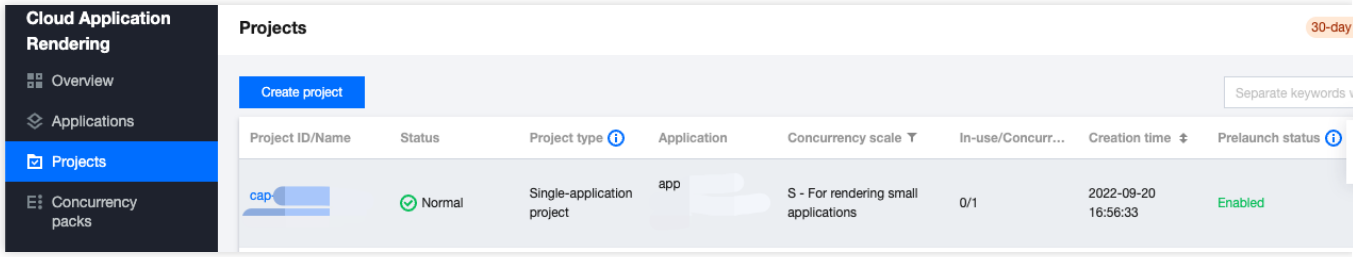
CpuUsage: If you notice a drop in FPS data, check if there is a CPU usage of 90-100%. If so, the current concurrency specification is insufficient. Try using a higher concurrency specification.

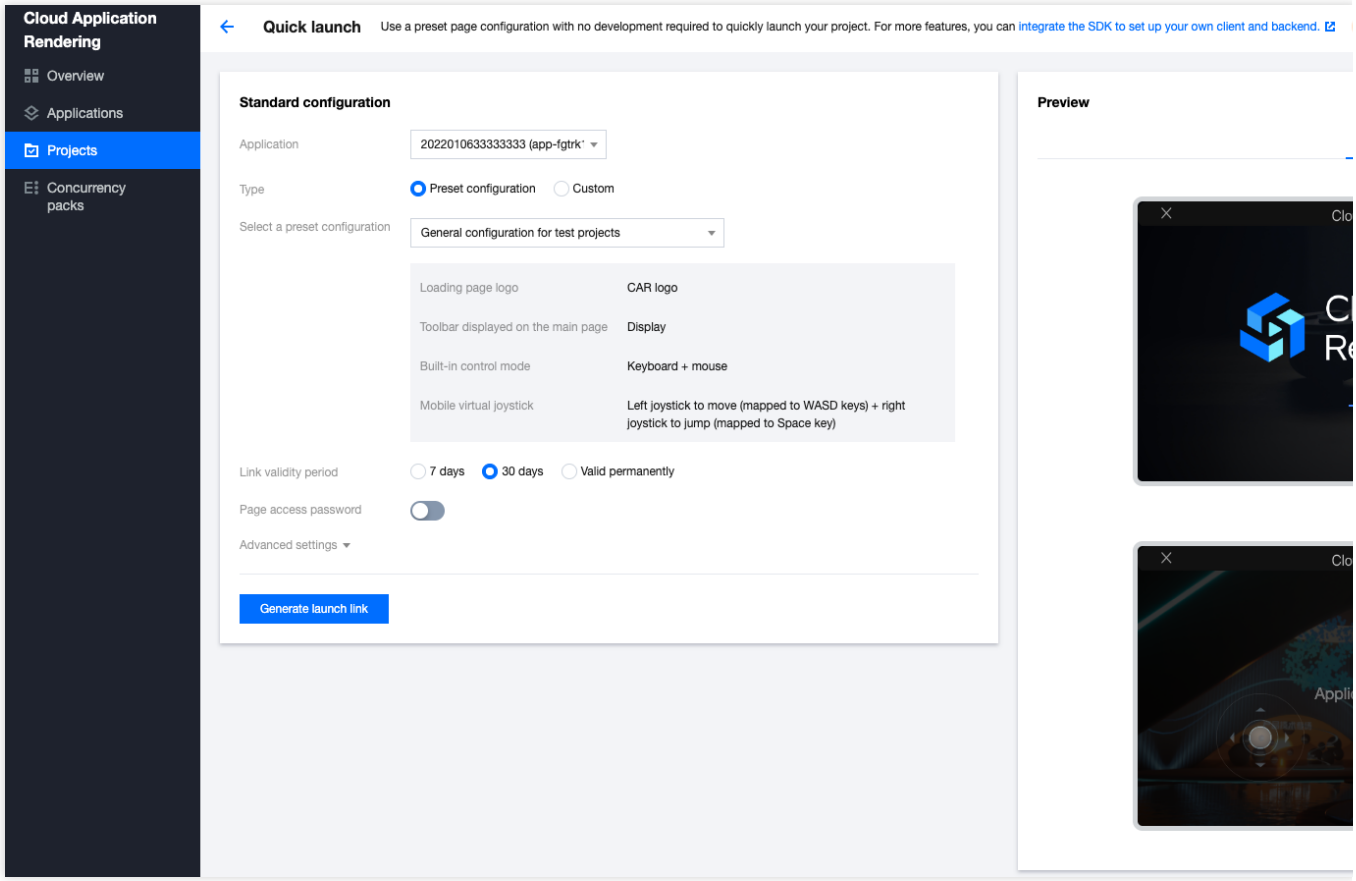
GpuUsage: Pay attention to the value of L (load). If you notice a drop in FPS data, check if there is an L value of 90-100%. If so, the current concurrency specification is insufficient. Try using a higher concurrency specification.
RequestId: If you have any questions, try to persist the connection and contact us with the `RequestId`. You can obtain the `RequestId` in the following methods:
If you use Chrome DevTools, you can obtain the `RequestId` of `CreateSession` on the **Network** tab.
If you have integrated to the CAR SDK, the returned values of the TencentCloud API include `RequestId`. We recommend you record it on the business backend.
Common issues encountered during testing:
A message is displayed indicating that there are no available concurrencies:
Check whether there are any available concurrencies under the project in the console. When a user exits the application, it takes about one minute for the concurrency to be automatically cleared. Only then will the concurrency become available so that a new user can connect to it.
The application cannot start or occurs a black screen:
We recommend you first check whether the path of the main executable file is correctly configured.
If the application uses window capturing mode, check whether the window title and class name are correctly entered. If they are configured incorrectly, a black screen will occur.
The application starts slowly:
Generally, the prelaunch feature is used to load an application in advance so the application will already be running when a user connects to it. Because multi-application projects do not support prelaunch, the applications under those projects take a longer time to start.
For a single-application project, you can enable prelaunch so that the application will be loaded in seconds when the user connects to it.
When a user exits the application, the concurrency they were connected to will be repossessed, cleared, and reset. When the concurrency becomes idle, the application will be prelaunched again. If the next user enters the application as soon as the concurrency becomes available, the application may have not been completely prelaunched.
The latency is high or the RTT value is abnormal: We recommend you first check for local network jitter. You can try accessing over a 4G/5G network.
The application is slow: This may be because your application has high requirements for the computing power but the concurrency specification is low (such as S). We recommend you try using a higher concurrency specification.
A message is displayed indicating that the connection is interrupted: If you are using a company Wi-Fi network, we recommend you check whether the network meets the CAR service environment, i.e., if all UDP ports are open. You can also check for local network jitter antry accessing over a 4G/5G network.
If your problem still persists, contact us for assistance. For more FAQs, see Cloud Application FAQs.

Quick Launch

Last updated：2024-01-26 12:00:38

After completing a test, you can launch a web page to access your cloud application using a preset page configuration with **no coding required**. To do so, click **Quick launch** in the **Operation** column to configure the front-end UI and generate a page link.





Directions

Step 1: Select the application you want to launch

A **single-application project** is only associated with one application, so there is no need to select the specific application to launch.
A **multi-application project** is associated with multiple applications, so you need to select the specific application you want to launch. If you want to launch multiple applications, you can select the first application to generate a link, and then select another application to generate another link. For more information, see Description of project types and How to implement concurrency sharing.

Step 2: Select a preset or custom configuration

The standard preset configuration allows you to launch your page quickly. You can modify the loading page logo, main page toolbar, built-in control mode, and virtual joystick for mobile devices.
Preset configuration: Three preset page configurations are available.
Custom Configuration: Use a custom configuration if you want your page to display a custom logo, display only the left joystick, or if you have other custom requirements.
If you have other special requirements, such as using a custom domain name, custom front-end display, or want to implement the Queue feature, you can integrate the SDK to develop your own front-end client and backend service. For a demo and detailed guides, see User guide.

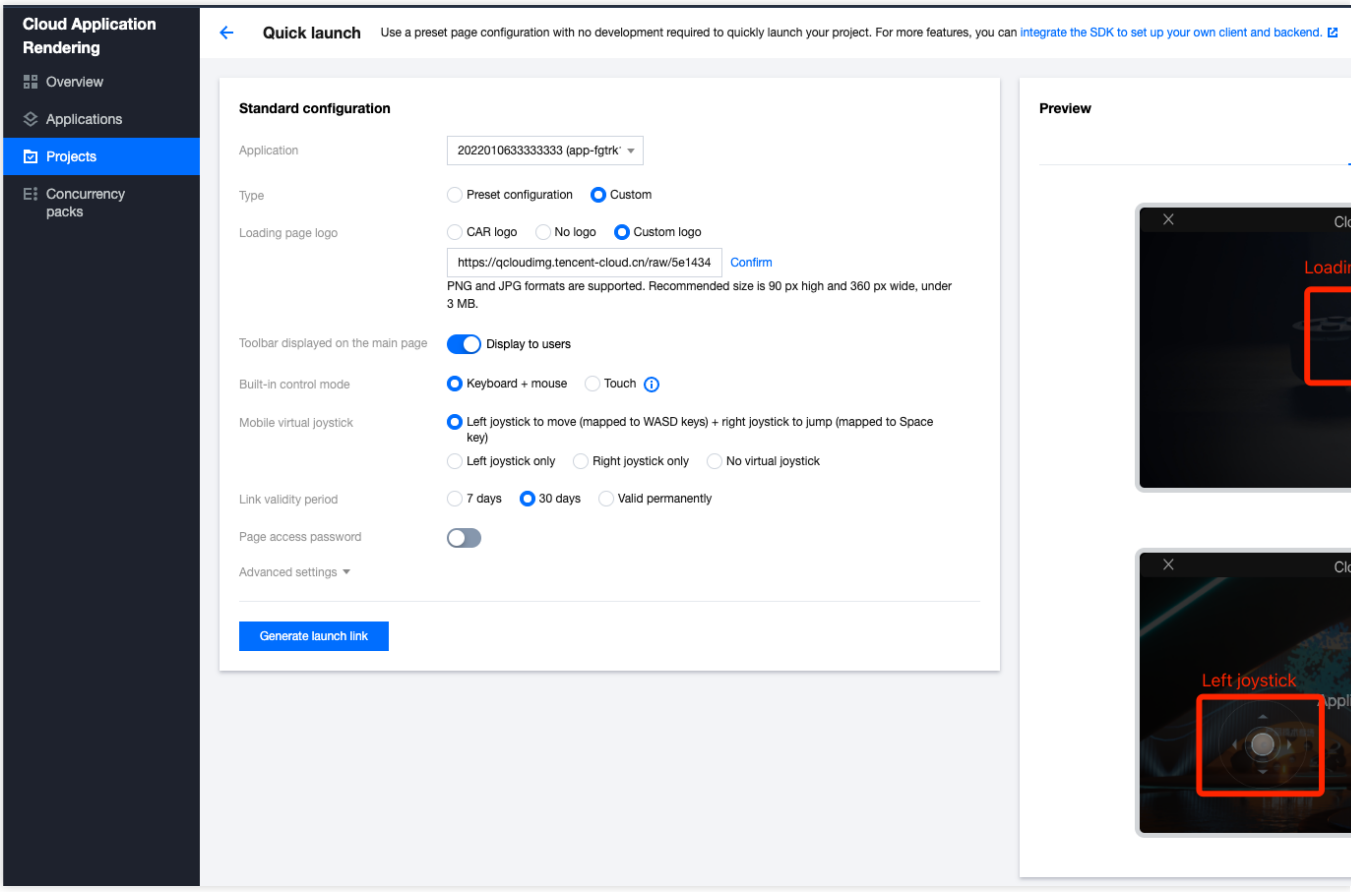
Preset configuration

	Description
General configuration for test projects	Displays frame rate and latency information, virtual buttons, debug panel, and other tools. Provided to help developers conveniently test the cloud application.
General configuration for mouse control applications	The Tencent Cloud logo and toolbar are hidden. **Keyboard and mouse events are sent to the cloud application**, and left/right virtual joysticks are enabled for users on mobile devices. **This is suitable for applications that only natively support keyboard and mouse controls. For users on mobile devices, a virtual joystick mapped to the WASD and Space keys needs to be displayed to allow the user to move around and jump.**
General configuration for multi-touch applications	The Tencent Cloud logo and toolbar are hidden. **Touch events are sent to the cloud application**, and mobile devices do not use the CAR virtual joystick plugin. **Note:** **This is only suitable for applications that natively support touch controls.** The application must be capable of managing touch window messages in Windows, and it has been locally tested with a physical Windows touch screen device to ensure successful operation. It is **NOT advisable** to rely on mouse-to-touch conversion within UE or Unity engines for testing touch message handling, as touch behavior on touch screen devices may behave unpredictably.

	If the application only natively supports mouse and keyboard commands, this configuration will not work for implementing touch controls. You can integrate the SDK and use the onTouchEvent callback to obtain the corresponding touch point coordinates and send keyboard/mouse commands as touch commands.

Custom configuration

	Description
Loading page logo	You can use a custom logo by entering the URL for the logo image, (for example, https://qcloudimg.tencent-cloud.cn/raw/5e1434b0dd32f26d9f064853aef02a33.png). The recommended logo dimensions are **90px (height) x 360px (width)**.
Toolbar displayed on the main page	**Mobile tools** include frame rate and latency information, full screen switch (only supports Android; iOS does not support full screen due to system limitations), rotate, mic switch, virtual joystick switch, virtual keyboard switch, debug panel, application restart, and application close. **PC tools** include frame rate and latency information, full screen switch, debug panel, application restart, and application close.
Application built-in control modes	**Keyboard + mouse**: Select if the application natively accepts keyboard and mouse commands to control the application. This typically requires the CAR virtual joystick plugin in order to accommodate users on mobile devices. **Touch**: Select if the application natively accepts touch commands for control and does not require the CAR virtual joystick plugin. The application must be capable of managing touch window messages in Windows, and it has been locally tested with a physical Windows touch screen device to ensure successful operation. It is **NOT advisable** to rely on mouse-to-touch conversion within UE or Unity engines for testing touch message handling, as touch behavior on touch screen devices may behave unpredictably. **How to implement multi-touch if the application only supports mouse and keyboard commands?** You need to integrate the SDK and use the onTouchEvent callback to obtain the corresponding touch point coordinates and send keyboard/mouse commands as touch commands.
Mobile virtual joystick	**Left joystick**: Sends keyboard WASD commands to the cloud application, usually used to control left/right/up/down movements. **Right joystick**: Sends keyboard Space key commands to the cloud application, usually used to jump.



Advanced settings

	Description
Landscape mode (default for mobile devices)	When enabled, the cloud rendered image will be displayed in landscape mode by default on mobile devices. This is typically enabled for applications and games whose display width is greater than the display height.
Mic enabled by default	When enabled, sound from the end user's microphone will be sent to the cloud application by default (requires end-user device support, such as Chrome requiring permission to use a microphone). This is typically enabled for applications and games that feature voice communication, such as group conferences and interactive live streaming applications.

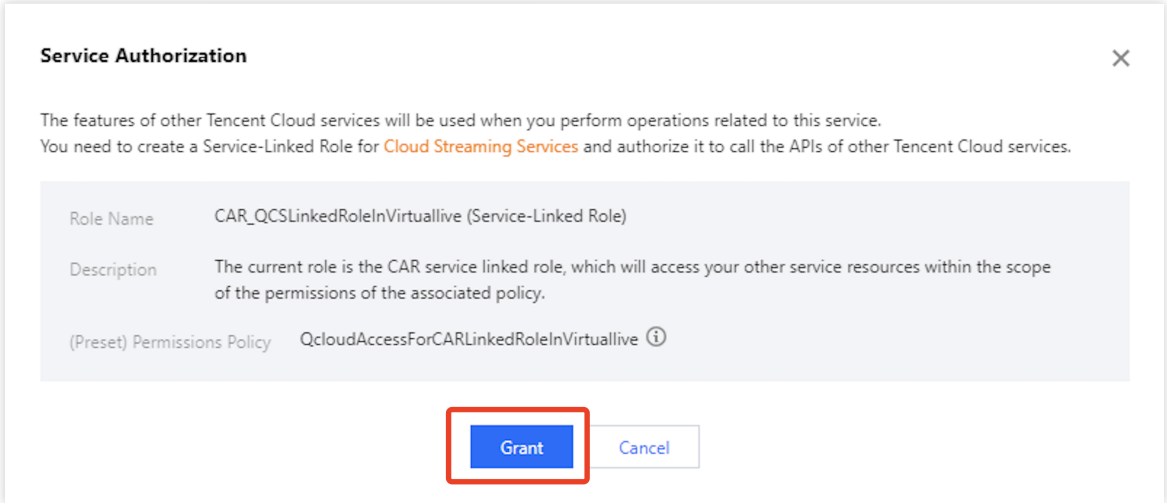
Stream Push Service
Pushing Streams to CSS

Last updated：2024-01-26 12:00:38

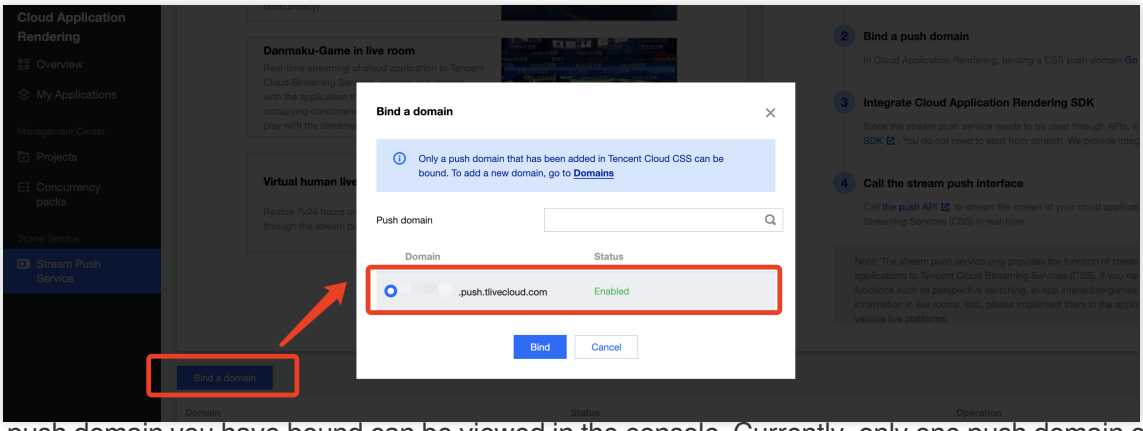
When running cloud applications for on-screen commenting interactive games, virtual meetings, and virtual exhibitions, it is often necessary to push a video stream to a live room at the same time so that more audience members in the live room can view the application. To do this, CAR provides the stream push service, which you can use with the following methods:
Push a cloud video stream to CSS by binding a push domain.
Push a cloud video stream to a third-party address according to the destination URL you enter. For more information, see Pushing Streams to Third-Party Addresses.
First, you need to **bind a push domain** so that you can call the stream push API on the running concurrency to push cloud video streams to CSS in real time.

Bind a push domain

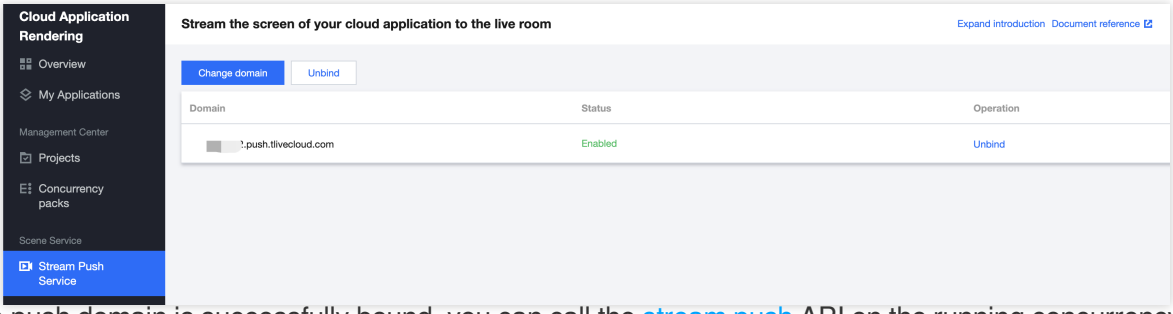
1. Go to the CAR console.
2. Click **Stream Push Service** on the left sidebar and select the Push to CSS page.
3. In the pop-up window, click **Confirm**. This operation authorizes the calling of CSS APIs, allowing you to use CSS and other related features.



4. Click **Bind a domain**, select a domain, and then click **Bind**. CSS comes with a default push domain. To add a new domain, go to the domain management page.



5. The push domain you have bound can be viewed in the console. Currently, only one push domain can be bound at a time. You can click **Change domain** to bind a different push domain.

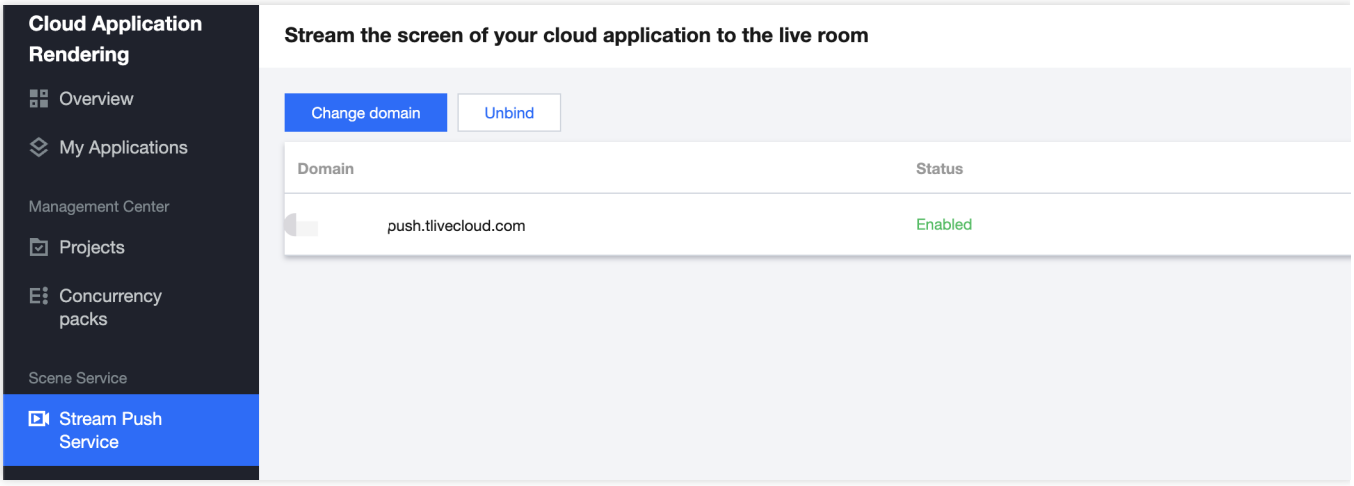


After a push domain is successfully bound, you can call the stream push API on the running concurrency to push cloud video streams in real time to CSS.

Unbind a push domain

You can unbind a push domain in the following steps.

1. Go to the CAR console.
2. Click **Stream Push Service** on the left sidebar and enter the **Push to CSS** page.
3. Click **Unbind**, and then confirm your operation in the pop-up window. After a push domain is unbound, the stream push API cannot be used for pushing video streams. If any push tasks are still running when the push domain is unbound, they will keep running until the concurrency is disconnected.



Pushing Streams to Third-Party Addresses

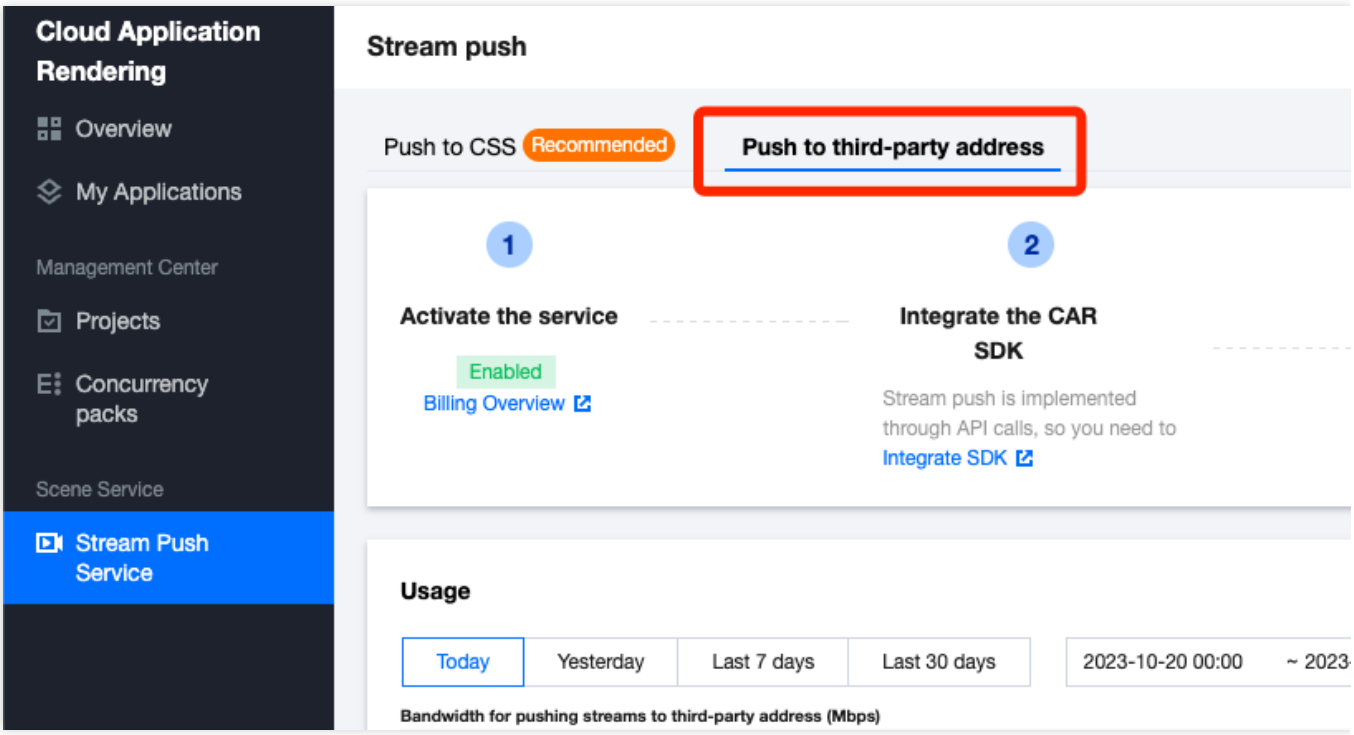
Last updated：2024-01-26 12:00:38

When running cloud applications for on-screen commenting interactive games, virtual meetings, and virtual exhibitions, it is often necessary to push a video stream to a live room at the same time so that more audience members in the live room can view the application. To do this, CAR provides the stream push service, which you can use with the following methods:
1) Push a cloud video stream to CSS by binding a push domain. For more information, see Pushing Streams to CSS.
2) Push a cloud video stream to a third-party address according to the destination URL you enter.

After you activate pushing streams to third-party addresses, call the stream push API on the running concurrency to push cloud video streams to third-party addresses you specify in real time.

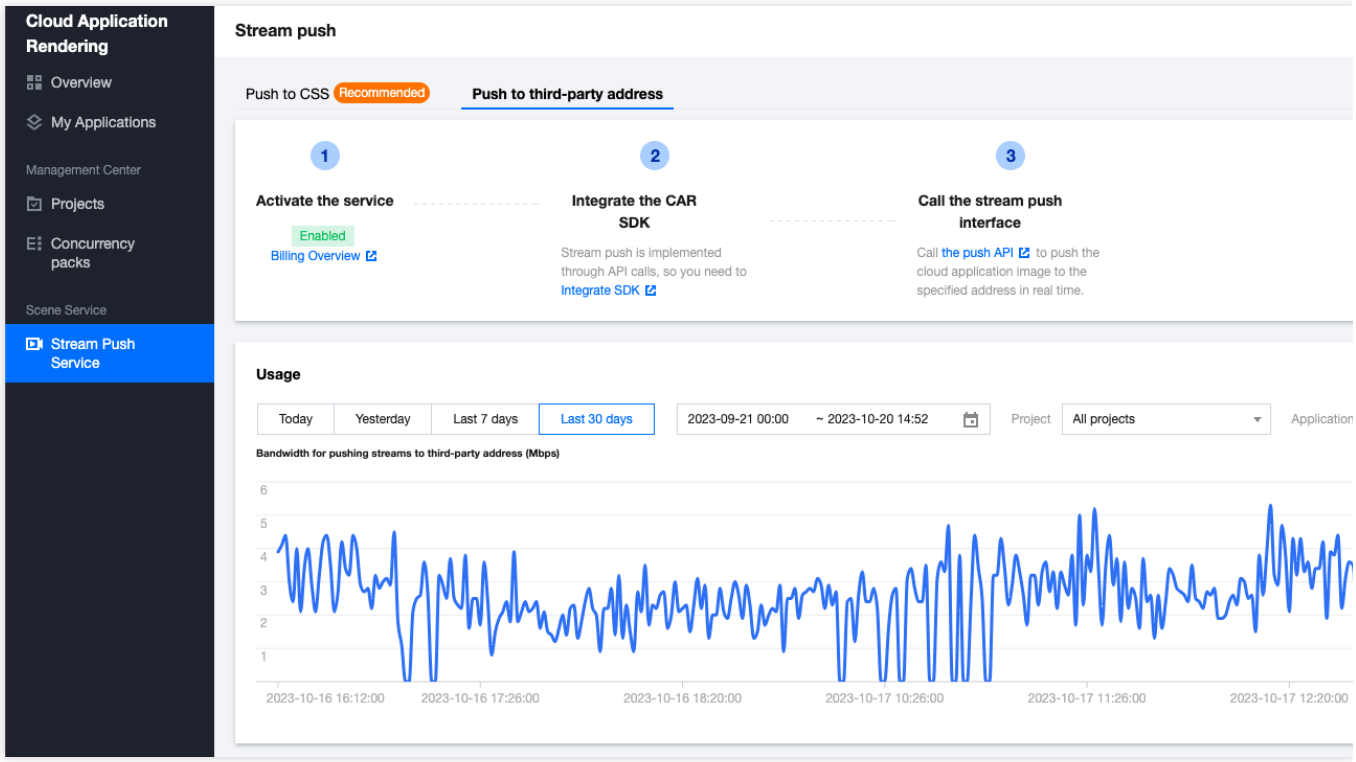
Directions

1. First, activate the service on the **Push to third-party address** page.



Note:
Using this feature will incur stream push bandwidth fees. For details, see Billing Overview.

2. After pushing streams to third-party addresses is activated, you can call the stream push API on the running concurrency to push cloud video streams to third-party addresses you specify in real time.
3. After using the stream push service, you can check the usage in the console.



Note:

The usage statistics may have a delay of 5-20 minutes.

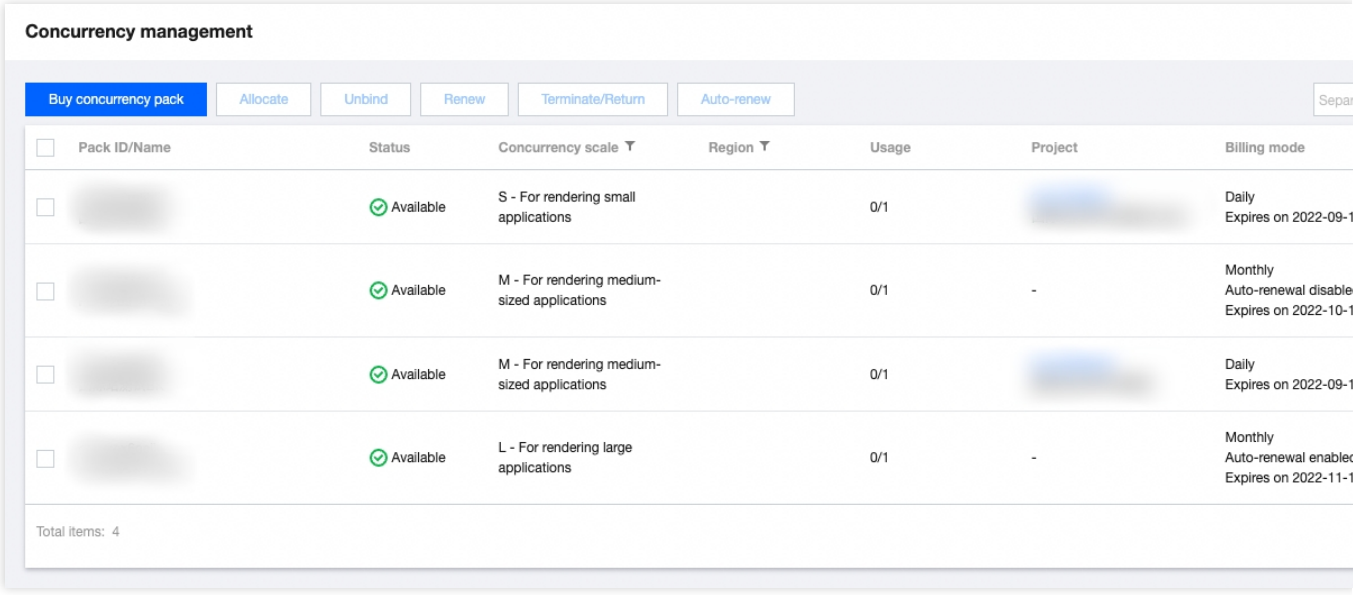
Concurrency Management
Purchasing a Concurrency Pack

Last updated：2024-01-26 12:00:38

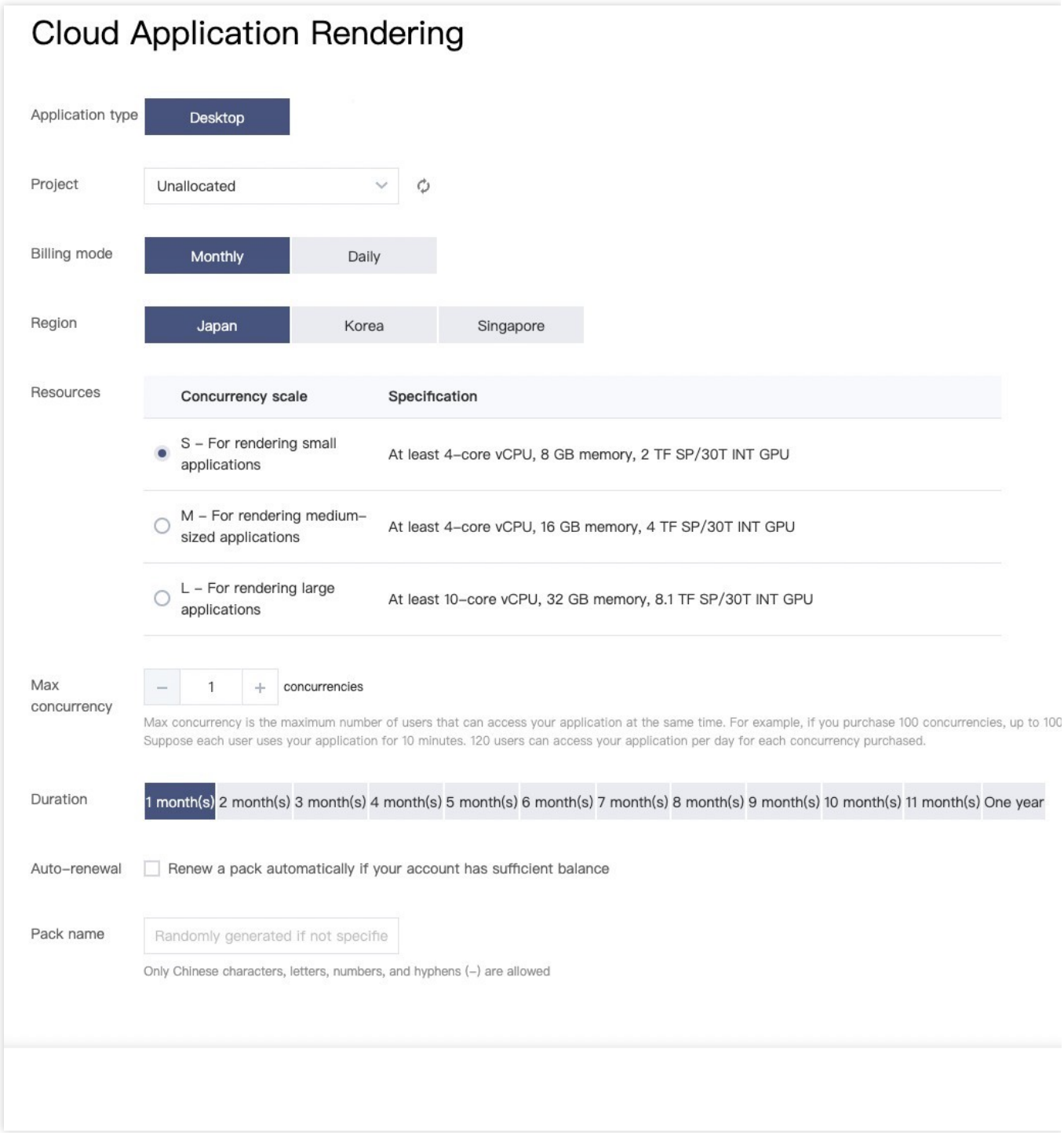
In CAR, users access your application by connecting to concurrencies. Each concurrency supports access by only one user at a time. If you want your business to sustain 100 concurrent online users, you need to purchase 100 concurrencies.
To purchase and manage concurrencies for your application, you need to purchase a concurrency pack. Concurrency packs are configured with resources and concurrencies, and can then allocated to your projects.

Directions

1. Go to the CAR console.
2. Click **Concurrency management** on the left sidebar and click **Buy concurrency pack** on the **Concurrency management** page.



3. On the **purchase page**, set the following configuration items:



Configuration Item	Description
Project	Select the project of the concurrency pack based on your business needs. You can select **To be allocated** first and allocate the concurrency pack on the **Project management** or **Concurrency management** page later.
Billing mode	Currently, CAR concurrency packs support two prepayment modes: monthly and daily subscriptions. For more information, see Billing Description.

Region	As cross-region CAR concurrency scheduling severely affects the operation experience, you need to select regions based where your business and users are located. Each concurrency can be used by only one user at a time. You need to select the number of concurrencies and the region based on the expected number of concurrent users in each region.
Concurrency scale	When you purchase a CAR concurrency pack, select a concurrency scale based on the concurrency scale that you specified for your project. If a project is selected, that project's concurrency scale will be selected.
Quantity	Select the number of concurrencies you want to purchase. You can purchase up to 100 concurrencies at a time.
Pack duration	Select the duration for the concurrency pack based on your business needs.
Auto-renew	If you enable **Auto-renew**, the concurrency pack will be automatically renewed monthly upon expiration if your Tencent Cloud account balance is sufficient.**Only monthly subscription supports automatic renewal**.
Pack name	Name the concurrency pack. If the pack contains multiple concurrencies, a number will be added to the end of the name of each CAR concurrency for identification.

4. Click **Buy Now** and confirm the information on the order details page.

Disconnect from user

Last updated：2024-01-26 12:00:38

In **Concurrency packs**, you can use the **Disconnect from user** feature to "remove" **all users** who are currently occupying the concurrency pack. For example, if 20 concurrencies in a concurrency pack is being used, this number will change to zero after the concurrency pack is disconnected from the users.

How do I fix the issue of abnormal concurrency usage?

If you find that a user is occupying a concurrency for a long time in a test, you can use the disconnection feature to remove the user.

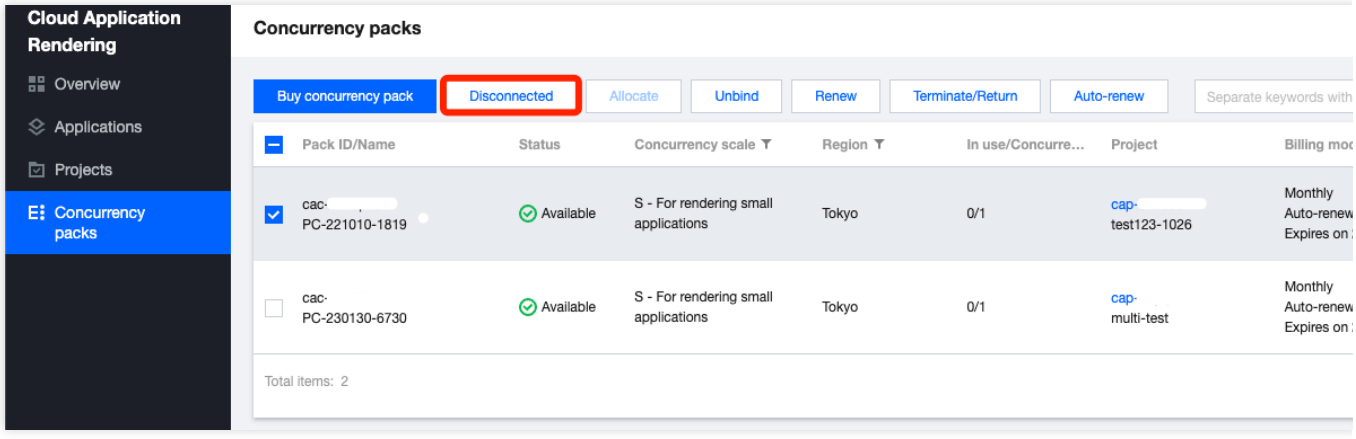
During technical integration, the SDK callback can be used to check whether a user is inactive for a long time. You can customize the idleThreshold parameter to set the threshold for user inactivity, so that the DestroySession API can be called to release the concurrency after the callback is received.

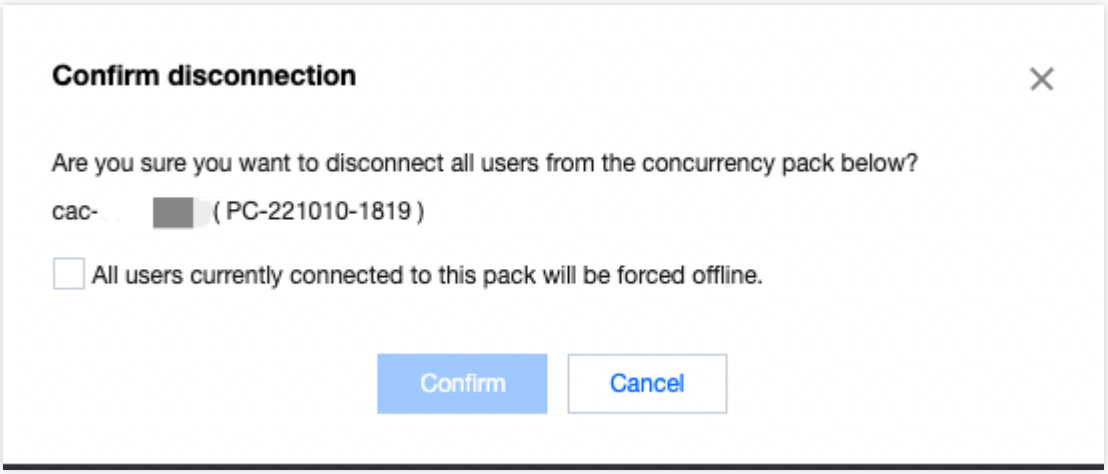
If the frontend user directly closes the page to close the connection to CAR, as DestroySession is not called, the CAR concurrency will wait 90 seconds for the user to reconnect. We recommend you maintain a heartbeat connection between the business backend and the client to check the user status and close the connection.

For more information, see Concurrency Scheduling.

Directions

1. Select the target concurrency pack.
2. Select **Disconnect from user** and click **Confirm**.



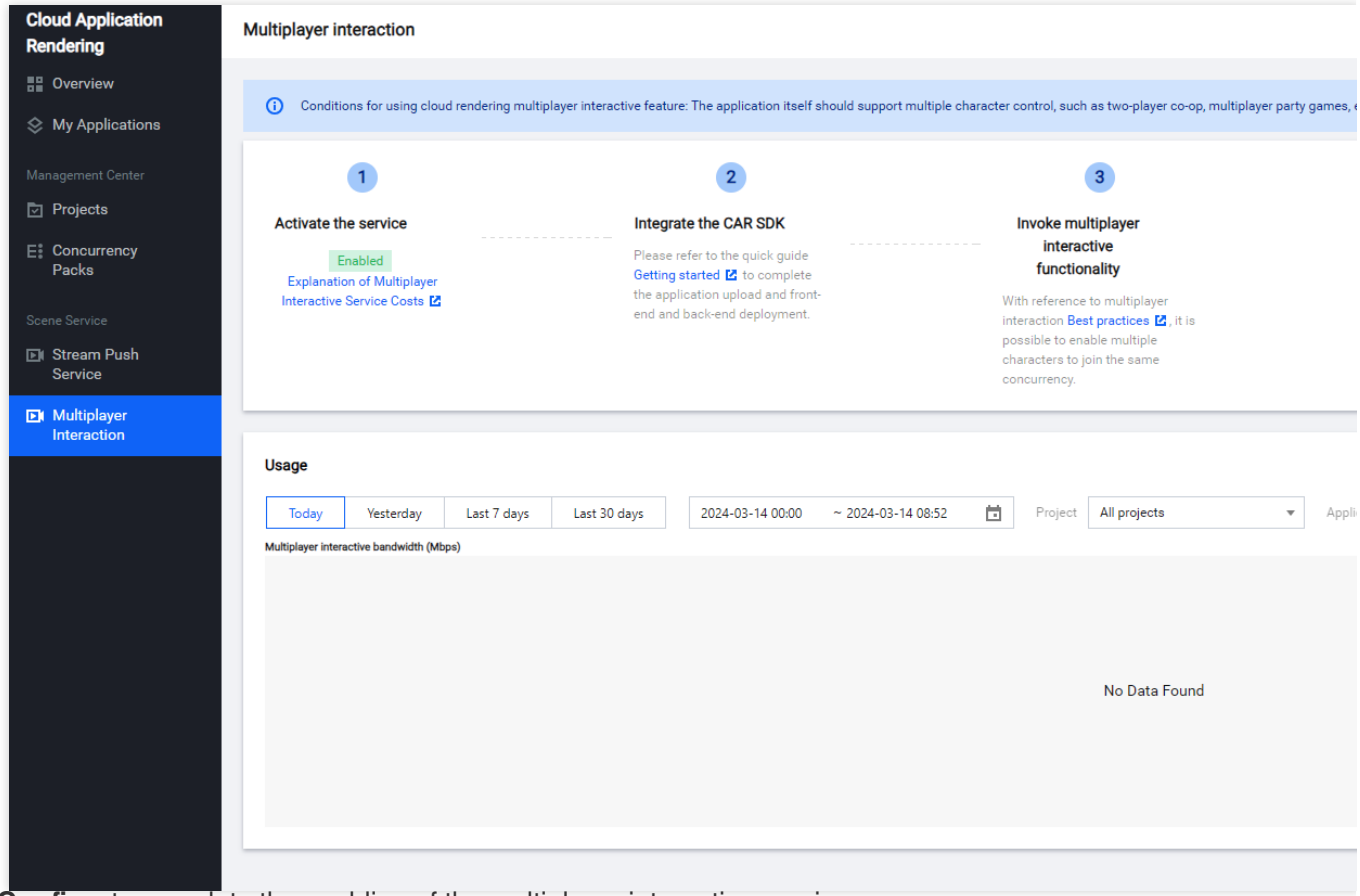


Multiplayer Interaction

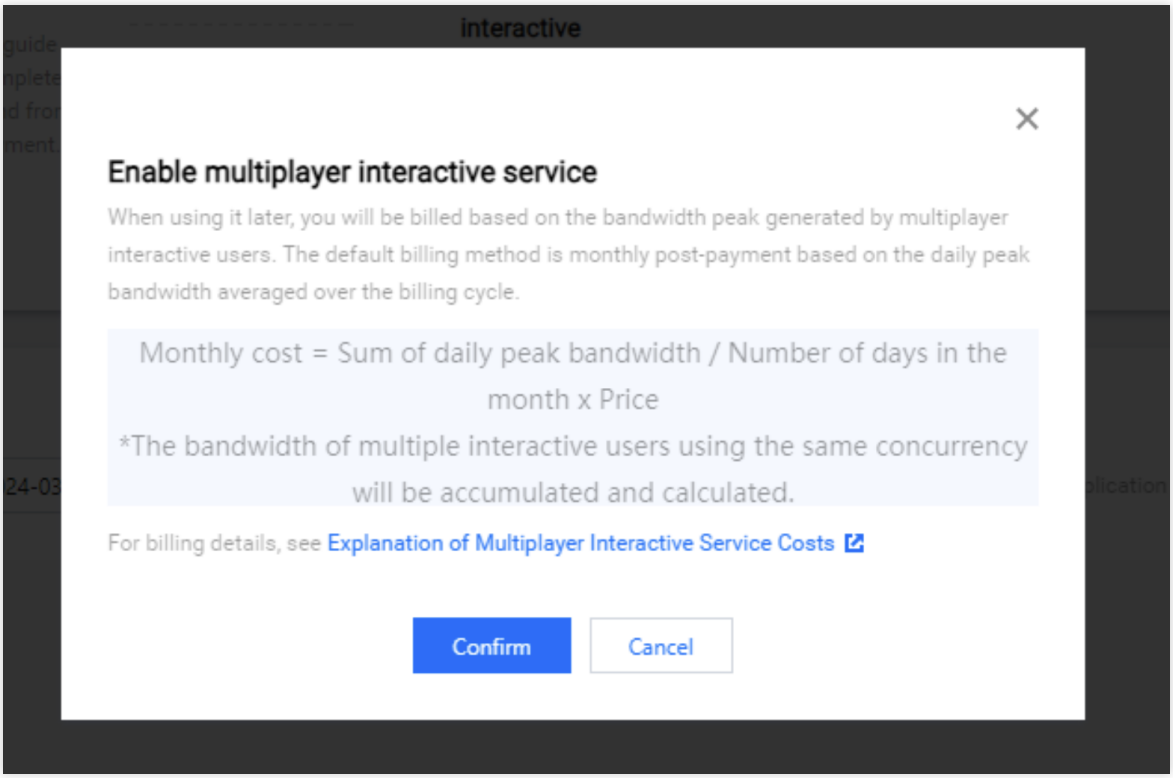
Last updated：2024-04-03 19:31:24

In the multiplayer interaction mode, the room is created by a CAR player (i.e., the room owner), and then other players (i.e., interactive audience) can join the same room via the room owner's UserId. In the same room, all users can see the same cloud-rendered scene through a cloud rendering connection. To use the multiplayer interaction feature, it must be actively enabled in the console, and then it can be called via API.

1. The multiplayer interactive feature requires activation via a whitelist. Before utilizing this function, please ensure that you have contacted the staff to enable whitelist access.
2. On the multiplayer interaction feature page in the console, confirm to enable the service.
Click **Enabled** to pop up the service confirmation dialog.



Click **Confirm** to complete the enabling of the multiplayer interaction service.



Note:

Using this feature will incur multiplayer interaction bandwidth fees. For details, please see billing overview.

3. After enabling the feature, refer to the document how to implement multiplayer interaction to use the multiplayer interaction feature.

4. Once a multiplayer interaction connection has occurred, it can viewed in the **Usage Statistics** from the console page.

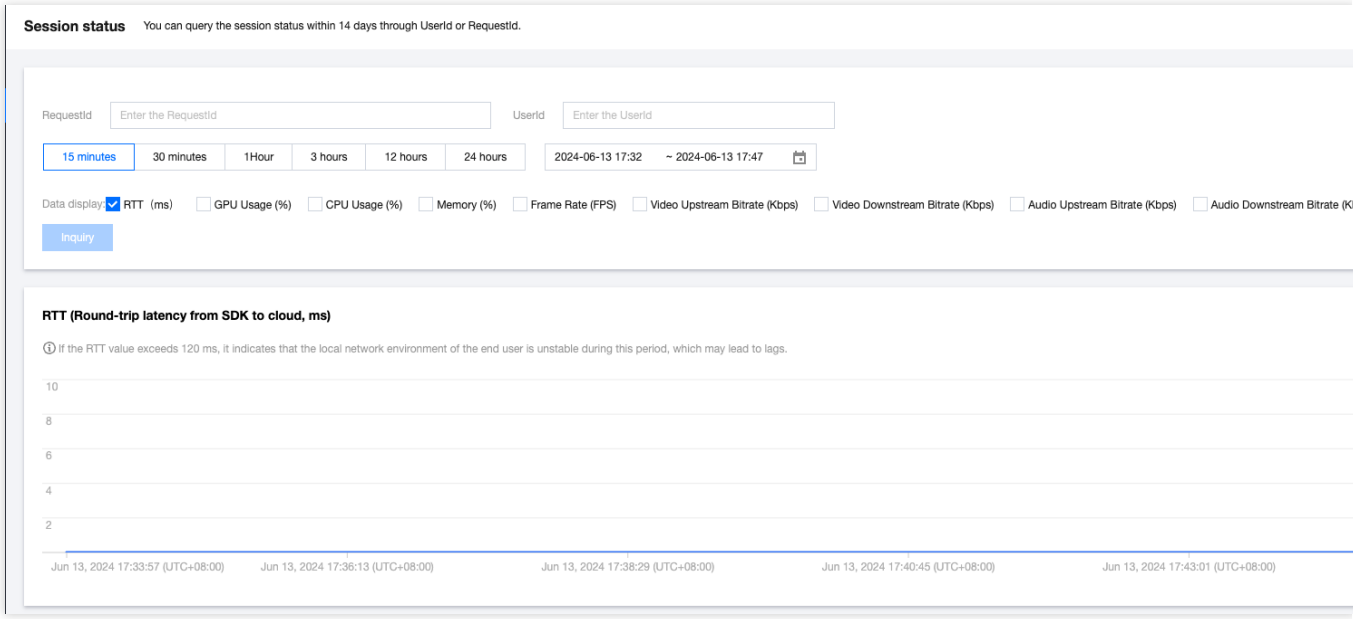
Note:

There will be a 5 to 20 minutes delay in accessing statistics.

Session Status

Last updated：2024-06-18 15:10:57

During access testing and online operations, if you encounter lag, application crashes, black screens, or blurred images, it is necessary to locate the issue by combining multiple data metrics. The console offers the session status query feature, which allows searching for session information within 14 days by entering the UserId or RequestId.



Data Metrics and Interpretations

RTT (ms):
Network latency metric Round-Trip Time (RTT) is an important parameter for measuring network performance. It represents the total time required for a data packet to travel from the sender to the receiver and back. If this metric is high or shows fluctuations, it indicates poor network conditions on the user's side.
GPU Usage (%):
If the GPU usage rate remains high during this connection, issues such as application lag, low rendering frame rates, and occasional application crashes may occur. It is recommended to switch to a concurrent model with a higher graphics card configuration or optimize the application to reduce GPU performance overhead.
CPU Usage (%):
If the CPU usage rate remains high during this connection, issues such as application lag, low rendering frame rates, and occasional application crashes may occur. It is recommended to switch to a concurrent model with more CPU cores or optimize the application to reduce CPU performance overhead.

Memory (%):
If the memory usage rate remains high during this connection, issues such as application lag and application crashes may occur. It is recommended to check for situations such as memory overflow and stack due to untimely task cleanup, and consider switching to a concurrent model with higher memory.
Frame Rate (FPS):
If the capture frame rate is too low, it may cause lag and other issues. This indicates that the current concurrency specification is insufficient to handle the load. It is recommended to use a higher specification of concurrency. When the capture frame rate is below 30 FPS, to stabilize the frame rate, Cloud Application Rendering (CAR) will insert duplicate frames to fill the encoding frame rate up to 30 FPS (without causing additional delay). A low frame rate is usually the result of other causes, and needs to be analyzed in conjunction with the CPU/GPU/memory usage rate at the same time, as well as application runtime logs.
Video Uplink Bitrate (Kbps)
Displays the bitrate for video uplink transmission. The CAR client SDK can collect data from the local camera and package it as input for the virtual camera on a cloud machine. If the local SDK device has the camera feature enabled, this data can be queried.
Video Downlink Bitrate (Kbps)
Displays the bitrate for cloud image transmission. If the bitrate is unstable or too low, it can cause the user to see a blurred image. This may be related to the user's local network conditions or the bitrate range settings of the project.
Audio Uplink Bitrate (Kbps)
Displays the bitrate for audio uplink transmission. The CAR client SDK can collect data from the local microphone and package it as input for the virtual microphone on a cloud machine. If the local SDK device has the microphone feature enabled, this data can be queried.
Audio Downlink Bitrate (Kbps)
Displays the bitrate for cloud audio transmission. This data metric is less likely to be affected by network conditions or performance.

Common Issues and Self-help Troubleshooting Methods

Users report severe lag during use. How should this be handled?

It is recommended to obtain the RequestId of this session from the backend data, and query the **RTT, GPU usage rate, CPU usage rate, memory, and frame rate** for that time period on the session status query page. Possible reasons for the lag include:
1. If the RTT is high, it can be determined that the user's poor network conditions are causing the screen to lag, along with a degree of packet loss leading to blurred images and dropped frames.
2. If the FPS is low, it can be determined that the issue is caused by the cloud machine's rendering results. You can check the GPU/CPU/memory usage rate to see whether there is a performance bottleneck, and it could also be due to the application itself.

Users report the screen goes black. How should this be handled?

First, based on the RequestId and the connection results of this session, determine whether a connection to CAR has been established. Sometimes, a black screen occurs because preliminary steps were not executed successfully, and the process has not progressed to connecting to the cloud display. If CAR has been connected, query the **RTT, GPU usage rate, CPU usage rate, memory, frame rate, and downlink bitrate** for that time period on the session status query page. Possible reasons for black screens include:
1. If the frame rate and bitrate are both 0, it can be determined that CAR has disconnected from the client. It is necessary to check the logs to see whether the session was terminated, and whether there was a reconnection logic in the code if the network instability led to disconnection.
2. If the frame rate and bitrate are not 0, it indicates that CAR is still connected. A black screen might be due to the application loading/starting, or because the application has frozen or crashed. The analysis should be done in conjunction with the **GPU usage rate, CPU usage rate, and memory** data, as well as application logs. Refer to Value-Added Features (Saving Application Logs and Archives) for methods to obtain cloud logs.

Users report blurred application images. How should this be handled?

First, obtain the RequestId of this session from the backend data, and query the **RTT, frame rate, and video downlink bitrate** for that time period on the session status query page. Possible reasons for blurred images include:
1. If the RTT is high and the downlink bitrate is unstable, it can be determined that the user's poor network conditions are causing the screen to lag, along with a degree of packet loss leading to blurred images and dropped frames.
2. If the downlink bitrate remains at a very low level, it is necessary to check the bitrate range settings in the project configuration. If the bitrate is set within a very low range (for example, below 3 Mbps), blurred images and mosaic effects are likely to occur in 3D scenes and complex images.