

Stream Compute Service

Getting Started

Product Documentation



Copyright Notice

©2013-2024 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

Getting Started

- Preparations

- Creating a Private Cluster

- Creating a SQL Job

- Creating a JAR Job

- Creating an ETL Job

- Creating a Python Job

Getting Started

Preparations

Last updated : 2023-11-08 10:51:37

Stream Compute Service is a streaming data aggregation and computing service based on the cloud. It allows you to easily connect to cloud data sources and access a complete set of supporting services without having to build or maintain infrastructure by yourself.

This document shows you how to get started with Stream Compute Service.

Step 1. Log in to the console

1. Log in to the [Stream Compute Service console](#).

Step 2. Create a cluster

Select [Compute resources](#) on the left sidebar. Select a region in the top left of the page and click **Create cluster**. Fill in the required information to [create a cluster](#).

After the cluster is successfully created, click **Associate space** in the **Operation** column to associate the cluster with a workspace so that the workspace can access the cluster's compute resources.

Step 3. Create a job

Select [Workspaces](#) on the left sidebar. Find the workspace associated and click **Go to space**.

Click [Jobs](#) on the left menu, select a region, click **Create**, and then click **Create job**.

Select a job type, enter a job name, select a cluster for the job, and click **Confirm**.

Next steps

You can create four types of jobs. For details, see [Creating a SQL Job](#), [Creating a JAR Job](#), [Creating a Python Job](#), and [Creating an ETL Job](#).

After a job is successfully created, you can view the [job type](#) and other [information](#) of the job and perform a series of [actions](#) on the job.

For information about job development, see [Job Development](#).

If you encounter a permission-related issue or want to manage the permissions of sub-accounts, see [Managing Permissions](#).

Creating a Private Cluster

Last updated : 2023-11-08 10:56:31

Prerequisites

1. You have a Tencent Cloud account. To learn about how to create an account, see [Signing Up](#).
2. If you log in with a sub-account, you need to grant the sub-account necessary permissions. For details, see [Granting access to Stream Compute Service](#) and [Granting access to other services](#).

Directions

Log in to the [Stream Compute Service console](#). Select **Compute resources** on the left sidebar and click **Create cluster**. Fill in the information and click **Buy now**. The cluster will be created successfully after you make the payment. In the cluster list, you will find the newly created cluster. After the status of the cluster changes from "Initializing" to "Running", you can start creating jobs with the cluster.

Select a region and availability zone. **Note that the cluster must be in the same region and availability zone as the resources (such as message queues and cloud databases) to be used by the cluster.**

Name and describe the cluster.

Select the number of CUs (minimum 12 and maximum 300). If you want to purchase more than 300 CUs, please [contact us](#).

Select a VPC and subnet. Please select the same VPC for resources (such as message queues and cloud databases) to be used by the cluster. The cluster will use an ENI and a certain number of IP addresses in the selected VPC. The subnet you select should have more CUs than the number you purchase.

Select a COS bucket. A private cluster stores data such as job checkpoints and JAR packages in COS buckets. If you don't have a COS bucket in the selected region, you can create one in the [COS console](#). Using COS buckets to store data will incur fees. For details, see [COS > Billing Overview](#).

Set the Flink UI password for accessing jobs. The username is "admin" by default.

You can tag the cluster. Tags allow you to manage your resources from different dimensions. For details, see [Tags](#).

You can also select **Auto-renewal** to avoid service suspension caused by expiration.

Note

To make the most of the underlying hardware resources, we recommend you purchase a multiple of six CUs. For performance considerations, we do not recommend you purchase more than 800 CUs for a cluster.

Core versions

Currently, new clusters only support Flink 1.13, Flink 1.14, and Flink 1.16 tasks. To create Flink 1.11 tasks, please [submit a ticket](#).

For existing clusters, only versions of already created tasks are supported. You can [submit a ticket](#) to upgrade a cluster.

By default, Flink 1.13, Flink 1.14, and Flink 1.16 tasks are deployed in application mode.

Creating a SQL Job

Last updated : 2023-11-08 10:52:34

Prerequisites

SQL jobs must be executed in a private cluster. If you don't have one yet, see [Creating a Private Cluster](#).

Step 1. Create a SQL job

Log in to the [Stream Compute Service console](#). Enter a workspace and click **Jobs** on the left menu. On the **Jobs** page, click **Create job**, select **SQL job** as the type, enter a job name, select a running cluster (where the job will be executed), and click **Confirm**.

Step 2. Grant access to related services

Select the job created and click **Development & Testing**. If you haven't granted Stream Compute Service access to related services, in the window that pops up, click **Authorize now** to allow Stream Compute Service to access resources such as message queues and cloud databases.

Step 3. Write SQL statements

After the authorization is completed, you can enter SQL statements in the code editing box to create a job quickly without preparing additional data. The example below performs the following actions:

1. Use the built-in connector Datagen to create the data source table "Data_Input", which includes the fields "age" and "score". The data type of both is BigInt. Datagen is a local data source of the cluster and can generate random data continuously.
2. Use the built-in connector Blackhole to create the data output table "Data_Output", which includes the fields "avg_age" and "avg_score". The data type of both is BigInt. Blackhole is a local data sink of the cluster and can receive data continuously.
3. Calculate the average of the "age" and "score" values in "Data_Input" and store the results in "Data_Output".

To use the data sources or sinks of other connectors, such as CKafka or Elasticsearch, see [Connectors](#). Please note that you will need to prepare your own data.



```
CREATE TABLE `Data_Input` ( --Step 1. Create the data source "Data_Input"
  age BIGINT,
  score BIGINT
) WITH (
  'connector' = 'datagen',
  'rows-per-second'='100', -- The number of data records generated per second

  'fields.age.kind'='random', -- A random number without range
  'fields.f_random.min'='1', -- The minimum random number
  'fields.f_random.max'='100', -- The maximum random number
```

```
'fields.score.kind'='random',      -- A random number without range
'fields.score.min'='1',            -- The minimum random number
'fields.score.max'='1000'         -- The maximum random number
);

CREATE TABLE `Data_Output` ( -- Step 2. Create the output data table "Data_Output"
  `avg_age` BIGINT,
  `avg_score` BIGINT
) WITH (
  'connector' = 'blackhole'
);

INSERT INTO `Data_Output` -- Step 3. Calculate the average of the "age" and "score"
SELECT AVG(age), AVG(score) FROM `Data_Input`;
```

Step 4. Configure job parameters

In **Job parameters**, configure parameters including the checkpoint interval and default operator parallelism. If you use other sources and sinks, you also need to select the corresponding connectors.

Step 5. Publish and run the SQL job

Click **Publish draft** to run and check the job and then click **Confirm** to publish the job. A version of the job will be created, and a version number will be automatically generated.

Under **Manage version**, you can view and switch to different versions of the job.

Select the version you want to run, click **Run version**, and then click **Confirm** to run the job.

Step 6. View job execution details

After you click **Confirm**, the job status will become "Operating" first and change to "Running" after the job is successfully run. After the job is successfully run, you can view its execution details via **Monitoring**, **Log**, and **Flink UI**.

Creating a JAR Job

Last updated : 2023-11-08 10:24:08

Prerequisites

JAR jobs must be executed in a private cluster. If you don't have one yet, see [Creating a Private Cluster](#).

Step 1. Prepare your JAR package

To create a JAR job, you need to develop your own JAR package. You can also [download](#) our word count example JAR package (flink-hello-world-1.0.0.jar).

Step 2. Upload the JAR package

Log in to the [Stream Compute Service console](#). Select **Dependencies** on the left sidebar and click **Create dependency**. Select the same region as the private cluster. For the upload method, **Local** is selected by default. Click **Select** and, in the window that pops up, select and upload the JAR package in step 1. Enter a package name, package description, and version description and click **Confirm**. The package will appear in the package list.

Step 3. Create a JAR job

Log in to the [Stream Compute Service console](#). Enter a workspace and click **Jobs** on the left menu. On the **Jobs** page, click **Create job**, select **JAR job** as the type, enter a job name, select a running cluster (where the job will be executed), and click **Confirm**.

Step 4. Grant access to related services

Select the job created and click **Development & Testing**. If you haven't granted Stream Compute Service access to related services, in the window that pops up, click **Authorize now** to allow Stream Compute Service to access services such as CKafka and TencentDB. For detailed directions, see [Granting access to other services](#).

Step 5. Configure the JAR job

In **Development & Testing**, from the drop-down list of **Main package**, select the package you uploaded. In **Main class**, type `com.tencent.cloud.oceanus.wordcount.WordCount`. You can click **Job parameters** to modify the default operator parallelism and add external dependencies.

Step 6. Publish and run the JAR job

1. Click **Publish draft** to run and check the job and then click **Confirm** to publish the job. A version of the job will be created, and a version number will be automatically generated.
2. Under **Manage version**, you can view and switch to different versions of the job.
3. Select the version you want to run, click **Run version**, and then click **Confirm** to run the job.



运行版本 V18 ×

不使用任何快照，直接运行

使用历史快照运行

选择快照 ▼

确定 取消

Step 7. View job execution details

After you click **Confirm**, the job status will become "Operating" first and change to "Running" after the job is successfully run. After the job is successfully run, you can view its execution details via **Monitoring**, **Log**, and **Flink UI**.

Creating an ETL Job

Last updated : 2023-11-08 10:51:17

Prerequisites

ETL jobs must be executed in a private cluster. If you don't have one yet, see [Creating a Private Cluster](#).

Step 1. Create an ETL job

Log in to the [Stream Compute Service console](#). Enter a workspace and click **Jobs** on the left menu. On the **Jobs** page, click **Create job**, select **ETL job** as the type, enter a job name, select a running cluster (where the job will be executed), and click **Confirm**.

Note

Currently, ETL jobs only support Flink 1.13. If your cluster does not support Flink 1.13, please [submit a ticket](#) to update it first.

Step 2. Grant access to related services

Select the job created and click **Development & Testing**. If you haven't granted Stream Compute Service access to related services, in the window that pops up, click **Authorize now** to allow Stream Compute Service to access resources such as message queues and cloud databases. For detailed directions, see [Granting access to other services](#).

Step 3. Configure the data source

Click **Job development > Draft** and drag the MySQL data source from the left to the canvas.

Click to configure the data source.

If you don't have an available database instance, click **Manage connections** and then click **Create connection** to add a new data source.

If you select **Tencent Cloud instance**, you can add a Tencent Cloud instance of the current account (which must have permission to query the corresponding Tencent Cloud product).

If you select **IP connection**, the IP address entered will be connected. Make sure the address is in the same VPC as the cluster or is connected with the cluster.

After selecting the data source, from the drop-down list of **Database/table**, choose a database and table to synchronize.

Click **Next**, select the fields to synchronize, and click **Confirm**.

Return to the canvas. You will see that the data source has been configured.

Step 4. Configure the sink

Drag a data sink from the left to the canvas and connect it to the data source.

Click the sink to configure the database instance and select the database/table.

Click **Next** and map the fields.

Click **Confirm**.

Step 5. Configure job parameters

Click **Job parameters** to configure the parameters. For details, see [Advanced Parameters](#).

Step 6. Publish and run the ETL job

Click **Run** to check and run the job.

Step 7. View logs

Click **Log** to view the logs of the job.

Creating a Python Job

Last updated : 2023-11-08 10:52:05

Prerequisites

Python jobs must be executed in a private cluster. If you don't have one yet, see [Creating a Private Cluster](#). Currently, Stream Compute Service supports running Python jobs developed using the open-source framework Flink 1.13 in environments installed with Python 3.7.

Step 1. Prepare your Python package

You need to develop your own Python package in order to create a Python job.

Step 2. Upload the Python package

Log in to the [Stream Compute Service console](#). Select **Dependencies** on the left sidebar and click **Create dependency**. Select the same region as the private cluster. For the upload method, **Local** is selected by default. Click **Select** and, in the window that pops up, select and upload the Python package in step 1. Enter a package name, package description, and version description and click **Confirm**. The package will appear in the package list.

Step 3. Create a Python job

Log in to the [Stream Compute Service console](#). Enter a workspace and click **Jobs** on the left menu. On the **Jobs** page, click **Create job**, select **Python job** as the type, enter a job name, select a running cluster (where the job will be executed), and click **Confirm**.

Step 4. Grant access to related services

Select the job created and click **Development & Testing**. If you haven't granted Stream Compute Service access to related services, in the window that pops up, click **Authorize now** to allow Stream Compute Service to access services such as CKafka and TencentDB.

Step 5. Configure the Python job

In **Development & Testing**, select the file you uploaded from the drop-down list of **Main package** and enter the entry class and class parameters. Stream Compute Service is currently built in with Python 3.7, which you can select as the **Python environment**. You can click **Job parameters** to modify the default operator parallelism and add external dependencies.

Step 6. Publish and run the Python job

Click **Publish draft** to run and check the job and then click **Confirm** to publish the job. A version of the job will be created, and a version number will be automatically generated.

Under **Manage version**, you can view and switch to different versions of the job.

Select the version you want to run, click **Run version**, and then click **Confirm** to run the job.

Step 7. View job execution details

After you click **Confirm**, the job status will become "Operating" first and change to "Running" after the job is successfully run. After the job is successfully run, you can view its execution details via **Monitoring**, **Log**, and **Flink UI**.