ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

Realtime Compute

User Guide

Product Version: V3.16.2

Document Version: 20220628

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.What is Realtime Compute? 06
2.Getting Started 07
2.1. Log on to the Realtime Compute console 07
2.2. Statistical analysis of frequently used words 07
2.2.1. Overview 07
2.2.2. Code development 08
2.2.3. Code debugging 09
2.2.4. Data O&M 11
2.3. Big screen service for Tmall Double 11 11
2.3.1. Overview 11
2.3.2. Scenario description 12
2.3.3. Preparations 12
2.3.4. Register a data store 13
2.3.5. Development 13
2.3.6. Operations and maintenance (O&M) 14
3.Manage projects 16
4.Data Storage 19
4.1. Overview 19
4.2. Authorize Realtime Compute to access a VPC 19
4.3. Overview of data storage 20
4.3.1. Overview 20
4.3.2. Types 21
4.3.3. Registration and usage 21
4.4. Register a DataHub data store 24
4.5. Register a Log Service data store 26
4.6. Register a Tablestore data store 27

4.7. Register an ApsaraDB RDS data store 28
5.Data Development 35
5.1. Create a job 35
5.2. Development 35
5.2.1. SQL code assistance 35
5.2.2. SQL code version management 36
5.2.3. Data store management 36
5.3. Debug job code 36
5.4. Publish a job SQL file 39
5.5. Start a job 40
5.6. Suspend a job 40
5.7. Terminate a job 41
5.8. View logs 42

1.What is Realtime Compute?

Realtime Compute is a big data processing platform that analyzes streaming data in real time based on Apsara Stack. You can use Alibaba Cloud Flink SQL to create streaming data analysis and computing jobs without the need to develop the underlying logic for streaming data processing.

As the demands for high data timeliness and operability increase, software systems need to process more data in less time. In traditional models for big data processing, online transaction processing (OLTP) and offline data analysis are separately performed at different times.

Realtime Compute is designed to meet the requirements for high timeliness of data processing. The business value of data decreases as time passes by. Therefore, after data is generated, the data must be computed and processed at the earliest opportunity. Traditional models for big data processing follow the scheduled processing mode, which accumulates and processes data in a computing cycle that can last hours or even days. The traditional models cannot meet the requirements for real-time data processing. Batch processing cannot meet the business requirements in delay-sensitive application fields, such as real-time big data analysis, risk management alerting, real-time forecast, and financial transactions. Realtime Compute processes data streams in real time. Realtime Compute reduces the data processing delay, implements real-time computing logic, and reduces computing costs. This helps you meet business requirements for real-time processing of big data.

Streaming data

Big data can be viewed as a series of discrete events. These discrete events form event streams or data streams in a timeline. Streaming data is continuously generated from thousands of data sources. In most cases, streaming data is sent in data records. Streaming data has a smaller scale than offline data. Each type of data is produced as a stream of events. Streaming data includes a wide variety of data, such as the log files that are generated by mobile or web applications, online purchases, in-game player activities, information from social networks, financial trade centers, geospatial services, and telemetry data from connected devices in data centers.

Realtime Compute provides the following benefits:

- Real-time and unbounded data streams

  Realtime Compute processes data streams in real time. Streaming data is continuously generated from data sources and is subscribed and consumed in chronological order. Data streams continuously flow into the Realtime Compute system. For example, when Realtime Compute processes data streams from website visit logs, the log data streams continuously enter the Realtime Compute system when the website is online. In Realtime Compute, unbounded data streams are processed in real time.

- Continuous and efficient computing

  Realtime Compute is an event-driven system in which unbounded events or data streams continuously trigger real-time computations. Each streaming data record triggers a computing task. Realtime Compute performs continuous and real-time computations over data streams.

- Real-time integration of streaming data

  Realtime Compute writes the computing result of each streaming data record into the destination data store in real time. For example, the system directly writes the data of a computed report to an ApsaraDB RDS instance for report display. Realtime Compute continuously writes the results into the destination data store in real time. Therefore, Realtime Compute can be viewed as a data source that generates data streams for the destination data store.

2.Getting Started

2.1. Log on to the Realtime Compute console

This topic describes how to log on to the Realtime Compute console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
2. Enter your username and password.

   Obtain the username and password that you can use to log on to the console from the operations administrator.

   > **Note** When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:
   >
   > - Uppercase or lowercase letters
   > - Digits
   > - Special characters, which include ! @ # $ %

3. Click **Log On**.
4. In the top navigation bar, move the pointer over Products and click **Realtime Compute**.
5. Specify **Organization** and **Region**.
6. Click **Blink**.

2.2. Statistical analysis of frequently used words

2.2.1. Overview

Statistical analysis of frequently used words is widely used in various scenarios, including the analysis of frequently used words in search engines, forums, and tags.

For example, you can easily view the latest and most frequently searched words on Sina Weibo based on real-time statistics. Statistical analysis of frequently used words is a simple word count job. In word count jobs for streaming data, real-time processing logic is used to analyze and display the frequently used words in real time.

If you are not familiar with big data computing, a word count job can help you easily get started. The word count job in big data computing is similar to a `Hello, World!` program that is often the first program that a developer learns to write. In the following topics, a word count job in Realtime Compute is used as an example to describe how to create a word count job based on the real-time processing logic. This example helps you understand the basic Flink SQL syntax and basic operations of Realtime Compute jobs. For example, you can create an SQL file for a job and publish the job.

2.2.2. Code development

This topic describes how to create a Realtime Compute job. In this example, a word count job is created.

Prerequisites

Before you create a word count job, a source table named stream_source and a result table named stream_result are created in external data stores. The stream_source table contains only one column. The column is named word and its data type is STRING. The stream_result table contains two columns. One column is named word and its data type is STRING. The other column is named cnt and its data type is BIGINT. The two tables are registered in Realtime Compute. A project is created. For more information, see Create a project.

1. Log on to the Realtime Compute console to go to the homepage of Realtime Compute.
2. In the top navigation bar, click **Development**.
3. Right-click the folder that you created.
4. Select **Create File**.
5. In the Create File dialog box, configure the following parameters:
   - File Name: Enter `wordcount`.
   - File Type: Select FLINK_STREAM / SQL.
   - Storage Path: Use the default setting.
6. Enter the following code in the code editor.

> **Note** In the SQL statements for the word count job, the STRING data type for the referenced table must be declared as the VARCHAR data type.

```
  create table stream_source (word varchar);
create table stream_result (word varchar, cnt bigint);
insert into
    stream_result
select
    t.word,
    count (1)
from
    stream_source t
group by
    t.word;
```

The following section describes the SQL code.

In line 1, the code is used to create a reference to the stream_source source table.

> **Note** Streaming data continuously enters Realtime Compute and triggers a stream processing procedure. Each streaming data record or each batch of data from the stream_source table triggers a stream processing procedure.

In line 2, the code is used to create a reference to the stream_result result table. The stream_result table stores the computing results of the word count job.

> **Note** Realtime Compute does not have built-in components for data storage. The result data is stored in external data stores, such as ApsaraDB RDS and Tablestore. This line of code is used to create a reference to a result table that contains the result data.

In lines 5 to 11, the computing logic is performed: Realtime Compute reads data from the stream_source table and counts the occurrence of words based on inbound data records.

> **Note** Flink SQL supports most standard SQL statements. This allows you to use Realtime Compute for stream processing in an efficient and cost-effective manner.

The method of performing a word count job for stream processing is similar to that for batch processing. The word count job for stream processing continuously processes unbounded data streams until the job is terminated.

2.2.3. Code debugging

Realtime Compute provides a powerful debugging feature to verify SQL statements. You can debug Realtime Compute jobs by simulating data stores where streaming data, static data, and result data are stored.

> **Note**
>
> - To prevent negative impacts on online data stores, Realtime Compute cannot read data from these data stores during the debugging process. Before debugging, you must prepare test data for input tables.
> - The output data of INSERT operations is exported only to local screens. This does not affect online systems.

Debugging method

1. In the upper part of the **Development** page, click **Debug**.
2. On the page that appears, click **Download Template** and modify the template based on your debugging rules.

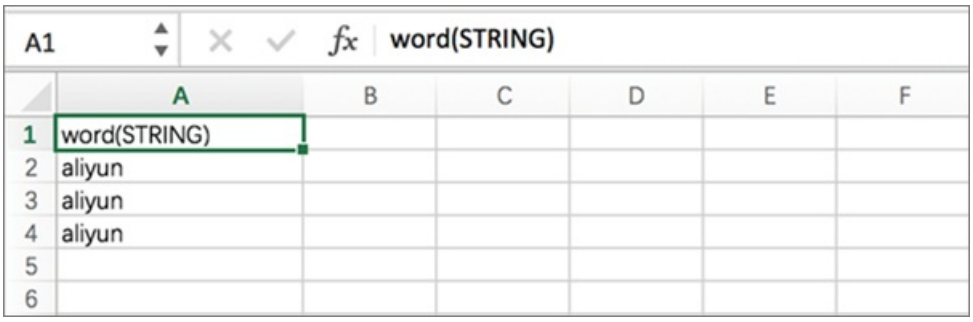
> **Note** The file that is uploaded for debugging must meet the following requirements:
>
> - The file size cannot exceed 1 MB, and the file can contain a maximum of 1,000 data records.
> - The file must be encoded in the UTF-8 format.
> - The file must be in the comma-separated values (CSV) format. Therefore, test data cannot contain commas (,).
> - Numeric values can be displayed only in the general format, and cannot be displayed in the scientific notation format.

3. Click **Upload** to upload the file.
4. In the dialog box that appears, click **OK**.
5. View the debugging result in the output window.

Sample file for debugging the word count job

> **Note** The file for debugging is in the CSV format. We recommend that you use the following software applications to open and modify the template:
>
> - Excel for Windows users.
> - Vim or Sublime Text for MacOS users. We recommend that you do not use Numbers because the software adds unnecessary fields when you modify CSV files.

Sample file for debugging the word count job



Test data

You can download test data and upload the data on the **Debug File** page.

> **Note** If you read this topic in a PDF file, you cannot download the *test data about frequently used words* by clicking the link. You can contact system administrators to obtain the test data.

View the debugging result

Stream processing of Realtime Compute is triggered by data streams. Each data record from the stream_source table triggers a stream processing task. After each task is complete, the computing result is exported. The test file contains three rows of data records. After each data record reaches Realtime Compute, a stream processing task is triggered. Therefore, a total of three data records are displayed. Realtime Compute uses the following computing logic:

- The data record aliyun in the first row reaches Realtime Compute. This is the first time that the system detects the word aliyun. Therefore, the computing result `<aliyun, 1>` is displayed.
- The data record aliyun in the second row reaches Realtime Compute. The system detects an existing record of `<aliyun, 1>` and increases the value by one. Therefore, the computing result `<aliyun, 2>` is displayed.
- The data record aliyun in the third row reaches Realtime Compute. The system detects an existing record of `<aliyun, 2>` and increases the value by one. Therefore, the computing result `<aliyun, 3>` is displayed.

The third computing result `<aliyun, 3>` is considered the final output of the debugging. Another sample of test data is provided for you to test the debugging feature. You can use different samples of test data and view the debugging result.

2.2.4. Data O&M

After the SQL file for your job passes the debugging test, you can click Publish in the upper part of the Development page to publish the job. Then, you can start the job on a Realtime Compute cluster on the **Administration** page.

Procedure

1. In the upper part of the **Development** page in the Realtime Compute console, click **Publish**. The **Publish New Version** dialog box appears.
2. In the **Resource Configuration** step, click **Next**.
3. In the **Check** step, click **Next**.
4. In the **Publish File** step, click **Publish**.
5. On the **Administration** page, view the word count job that you published.
6. Find the word count job and click **Start** in the Actions column.
7. In the Start dialog box, specify **Start Time of Reading Data** and click **OK**. Then, the job can be scheduled by the Realtime Compute cluster.

Result

After the job is started, you can click the name of the job and view the job information on the **Overview** tab.

- Q: Why does the word count job have no input or output when the job runs on the distributed Realtime Compute cluster?
- A: When you created the `my_source` and `my_result` tables, you did not specify the data storage type of the referenced data source. In this scenario, the source table is considered a random table of strings or numbers, and the result table is considered discarded data.

2.3. Big screen service for Tmall Double 11

2.3.1. Overview

During Double 11, a big screen shows the total sales of Alibaba Group in real time. The big screen service is a highlight for the shopping festival.

Stream processing for the big screen service was previously based on Apache Storm that is an open source distributed real-time computation system. The Apache Storm-based development process required approximately one month to complete. The application of Flink SQL shortened the development process of the big screen service to three days. The underlying layer of Realtime Compute removes the Apache Storm modules that are designed for execution optimization and troubleshooting. This implements a higher processing efficiency for Realtime Compute jobs.

2.3.2. Scenario description

The streaming data input for the Tmall big screen service is the transaction data from the Tmall platform. The incoming transaction data is organized based on a two-dimensional table: `tmall_trade_detail`.

Field	Type	Description
tid	BIGINT	The order ID.
buyer_uid	BIGINT	The buyer ID.
seller_uid	BIGINT	The seller ID.
gmtdate	TIMESTAMP	The time when the order is completed.
payment	DOUBLE	The order amount.

Realtime Compute calculates two metrics based on the preceding table: the total number of orders and the total order amount up to the current time. The two metrics are written to an online RDS system and displayed on a big screen in real time. The online RDS system is used to store the result table: `tmall_trade_state`.

Field	Type	Description
gmtdate	VARCHAR(16)	The date when the order is completed.
trade_count	BIGINT	The total number of orders.
trade_sum	DOUBLE	The total order amount.

The following topics describe how to build an end-to-end solution for the Tmall big screen service in around 10 minutes.

2.3.3. Preparations

Before you write Flink SQL statements for a Realtime Compute job, you must register data stores for source tables and result tables in Realtime Compute. This topic describes how to register a data store in Realtime Compute. In this topic, the data store is DataHub.

Create a DataHub topic

1. Log on to the DataHub console. For more information, see the "Log on to the DataHub console" section in DataHub User Guide.
2. On the Projects page, find the project that you want to manage and click **View** in the Actions column.
3. On the page that appears, click **Create Topic**.
4. Configure the topic based on the schema of the tmall_trade_state RDS table in the "Scenario description" section.

After you perform the preceding steps, you can write Flink SQL statements for your Realtime Compute job.

Upload data to DataHub

Log on to the **DataHub** console. Then, perform the following steps to upload data to the DataHub topic that you created.

1. Log on to the DataHub console.
2. In the left-side navigation pane, click **Data Acquisition**.
3. Click **Upload File**.
4. On the page that appears, double-click the project that you want to manage and click the DataHub topic to which you want to upload data.
5. Click **Select File** to select a file.
6. Click **Upload**.

To simplify the test procedure, you can use the test data about Double 11. You can download the data and then upload the data to the DataHub topic for data collection.

2.3.4. Register a data store

The data store registration feature of Realtime Compute allows you to easily register DataHub topics, create tables, and reference data stores. To register a data store, perform the following steps:

Procedure

1. Log on to the Realtime Compute console to go to the homepage of Realtime Compute.
2. In the top navigation bar, click Development.
3. In the left-side navigation pane, click the **Storage** tab.
4. Click the **DataHub Data Storage** folder.
5. On the top of the page, click **+Registration and Connection**.
6. Register a DataHub project in Realtime Compute. For more information about parameter settings, see Register a DataHub project.

   If you use ApsaraDB RDS for MySQL to store the result data for data visualization, you must register an ApsaraDB RDS data store in Realtime Compute. For more information, see Register an RDS instance.

2.3.5. Development

After the data has been collected to Realtime Compute, you can continue to edit Flink SQL statements.

1. Create a reference to the source.

   To create references to the DataHub source table and RDS result table, click Data Storage in the left-side navigation pane of the **Development** page in the Realtime Compute console, and perform the following operations:

   - Find the target DataHub topic, and click **Reference as Source Table**. Realtime Compute automatically parses the schema of the topic and adds the corresponding SQL statements to the **Development** page.
   - Find the target RDS table, and click **Reference as Result Table**. Realtime Compute automatically parses the schema of the table and adds the corresponding SQL statements to the **Development** page.

2. Edit Flink SQL statements.

   If you have created the DataHub topic and RDS table as described in the previous topics, the Flink SQL code for the tmall_d11 job can be executed directly. Otherwise, change the names of the DataHub topic and RDS table based on the topic and table that you have created. The sample code is as follows:

   ```
   replace into tmall_trade_state
       select
           from_unixtime(FLOOR(tmall_trade_detail.gmtdate/1000), 'yyyy-MM-dd') as gmt_date
   ,
           count(tid) as trade_count,
           sum(payment) as trade_sum
       from
           tmall_trade_detail
       group by
           from_unixtime(FLOOR(tmall_trade_detail.gmtdate/1000), 'yyyy-MM-dd');
   ```

   > **Note** You can modify the information about tables and fields as required.

3. Debug the Flink SQL code.

   The data during the Double 11 Shopping Festival is available for testing. To debug the code, download the test data and upload the data on the **Development** page for debugging.

4. Publish the SQL file for the tmall_d11 job.

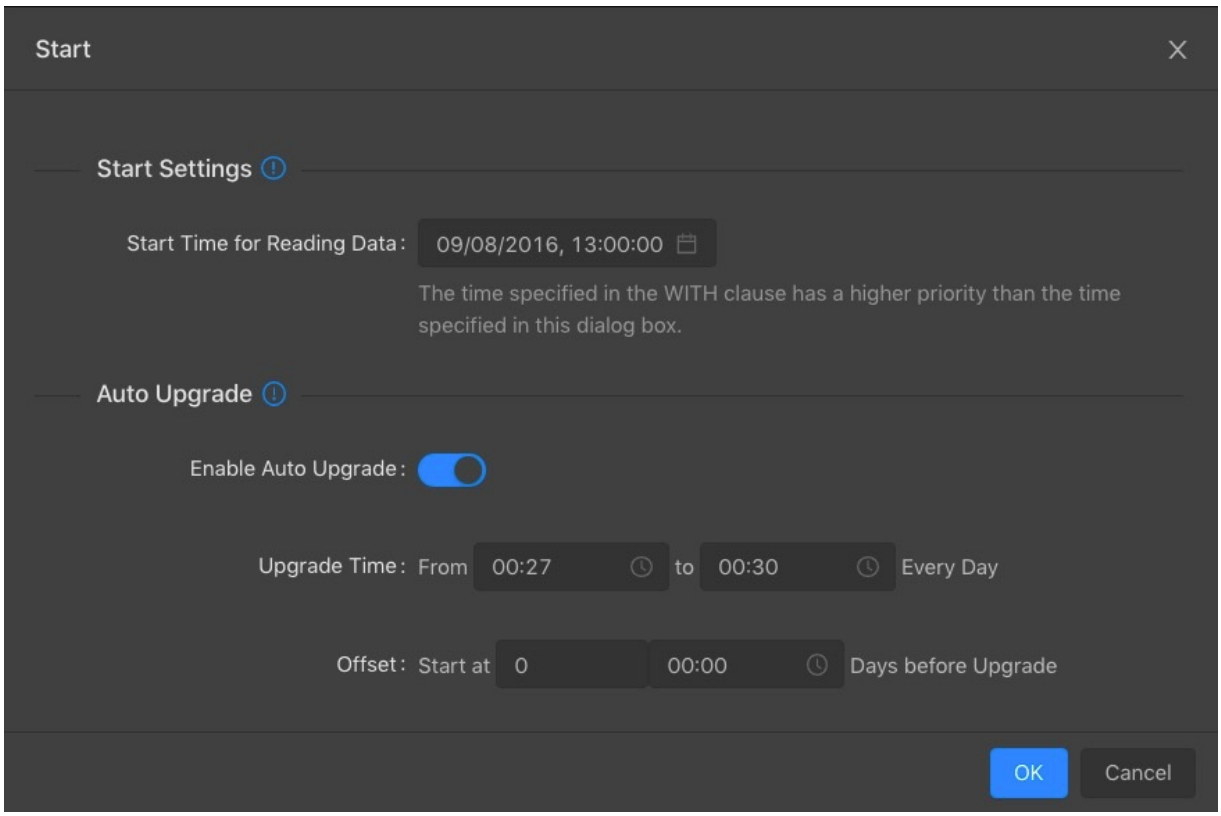
   After the computational logic has been verified in the debugging phase, click **Publish** on the **Development** page to publish the SQL file for the tmall_d11 job. Then, you can view the tmall_d11 job on the **Administration** page of the Realtime Compute console, and manage the job in the production environment, such as starting the job.

2.3.6. Operations and maintenance (O&M)

On the Administration page, you can click Start in the Actions column and specify the parameters on the page that appears to start a stream processing job, for example, the tmall_d11 job.

> **Note** After you click Start, a dialog box is displayed. In the dialog box, you can specify the start time for reading data from the source data store.

The specified start time must be earlier than the file upload time. For example, the start time can be one hour earlier than the file upload time. In the Double 11 use case, the current time is 14:10, and 10 minutes have elapsed since the source data is uploaded. Therefore, the start time is set to 13:00.



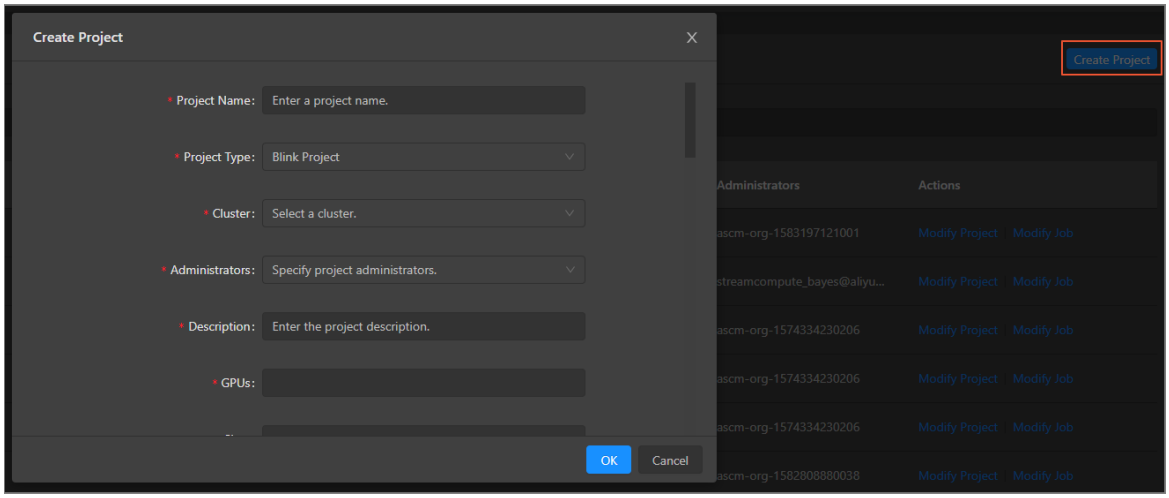
You can check the result data in the ApsaraDB RDS data store after the job runs as expected. In the result table, five transactions and a turnover of CNY 500 are displayed. This is consistent with the source data for testing. In this way, an end-to-end verification is performed to check the SQL code.

3.Manage projects

This topic describes how to create and search for a project.

Create a project

1. Log on to the Realtime Compute console.
2. In the top navigation bar, move the pointer over your profile picture and click **Project Management**.
3. In the upper-right corner of the **Projects** page, click **Create Project**.
4. Configure the parameters of the project.



Parameter description

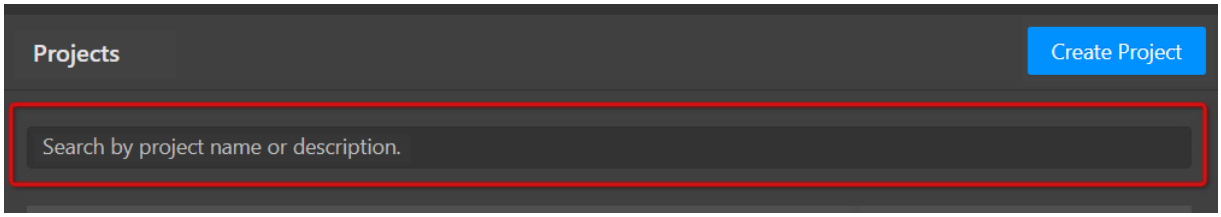
Parameter	Description
Project Name	The name of the project.
Project Type	The type of the project. By default, **Blink Project** is selected.
Cluster	The cluster on which the jobs in the project run.
Administrators	The administrator of the project.
Description	The description of the project.
GPUs	The number of GPUs that are used by the project.
Slots	The number of compute units (CUs) that are used by the project. One CU is assigned 1 CPU core and 4 GB of memory.
Alert Methods	The methods that are used to send alert notifications when the job runs abnormally. Alert notifications can be sent by using text messages or TradeManager messages.
File Types	The file types that are supported in the project. You can use the default setting.

Parameter	Description
Storage Types	The data store types that are supported in the project. You can use the default setting.
Max Data Stores	The maximum number of data stores that can be registered in Realtime Compute. You can use the default setting.
Max File Versions	The maximum number of code versions for an SQL file. You can use the default setting.
Max Folders	The maximum number of folders that can be created in the project. You can use the default setting.
Max Folder Levels	The maximum number of folder levels that can be created in the project. You can use the default setting.
Max Files	The maximum number of job SQL files that can be created in the project. You can use the default setting.
Max Resources	The maximum number of JAR files and DICTIONARY resources that can be uploaded. You can use the default setting.
Max Referenced Resources	The maximum number of JAR files and DICTIONARY resources that can be referenced. You can use the default setting.
Monitoring and Alerting	Specifies whether to enable the monitoring and alerting feature. You can use the default setting.
Data Collection	Specifies whether to collect data when a job is running. You can use the default setting.
Data Display	Specifies whether to display data. You can use the default setting.
Metastore	Specifies whether to display metadata. You can use the default setting.
Data Storage	Specifies whether to enable the data store registration feature. This feature is enabled by default. You can use the default setting.
Engine	Specifies whether to enable the data engine feature. You can use the default setting.
Online Logs	Specifies whether to record run logs of jobs. This feature is enabled by default. You can use the default setting.
Resource Management	Specifies whether resources such as JAR files can be uploaded. This feature is enabled by default. You can use the default setting.
Switch Version	Specifies whether the job version can be changed. This feature is enabled by default. You can use the default setting.
Project Protection	Specifies whether to enable the project locking feature. You can use the default setting.

5. Click **OK**.

Search for a project

On the **Projects** page, enter a keyword or the full name of a project in the search box to search for the project.



4.Data Storage

4.1. Overview

This chapter describes data storage systems supported by Realtime Compute.

4.2. Authorize Realtime Compute to access a VPC

Before you use Realtime Compute to access storage resources in a virtual private cloud (VPC), you must authorize Realtime Compute to access the VPC. This topic describes how to authorize Realtime Compute to access a VPC.

Procedure

1. Log on to the Realtime Compute console.
2. Move the pointer over the username in the upper-right corner.
3. In the list that appears, click **Project Management**.
4. In the left-side navigation pane, click **VPC Access Authorization**.
5. In the upper-right corner of the **VPC Access Authorization** page, click **Add Authorization**.
6. In the **Authorize StreamCompute VPC Access** dialog box, configure the parameters as required. The following table describes the parameters.

Parameter	Description
Name	The name of the VPC.
Region	The region where the storage resource resides.
VPC ID	The ID of the VPC. To view the VPC ID of an ApsaraDB RDS instance, perform the following steps: i. Log on to the ApsaraDB RDS console. ii. In the top navigation bar, select the region where the ApsaraDB RDS instance resides. iii. On the Instances page, find the ApsaraDB RDS instance and click the instance ID in the Instance ID/Name column. iv. In the left-side navigation pane, click **Database Connection**. v. On the **Instance Connection** tab, view the VPC ID next to the value of **Network Type** in the **Database Connection** section. For example, the VPC ID is `vpc-bp1lysht98wrvl9n3****`.

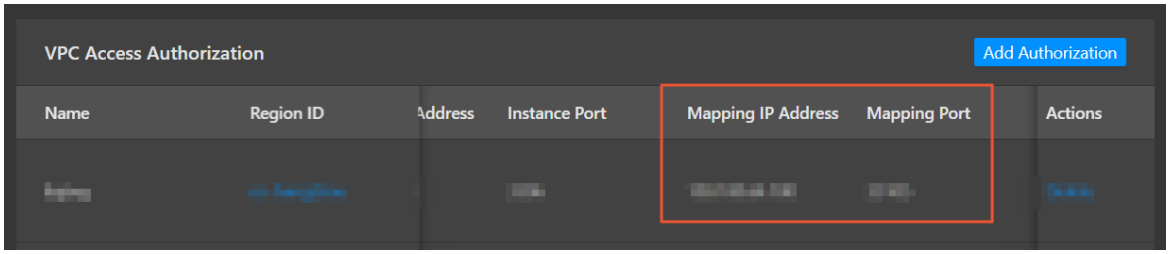
Parameter	Description
Instance ID	The instance ID of the storage resource in the VPC. To view the ID of an ApsaraDB RDS instance, perform the following steps: i. Log on to the ApsaraDB RDS console. ii. In the top navigation bar, select the region where the ApsaraDB RDS instance resides. iii. On the Instances page, find the ApsaraDB RDS instance and click the instance ID in the Instance ID/Name column to go to the **Basic Information** page. iv. View **Instance ID** in the **Basic Information** section.
Instance Port	The port of the storage resource in the VPC.

FAQ

Q: How do I set the url parameter when I use data definition language (DDL) statements to reference a storage resource in a VPC?

A: When you use DDL statements to reference storage resources in a VPC, you can set the url parameter in the WITH clause based on the **Mapping IP Address** and **Mapping Port** parameters on the **VPC Access Authorization** page. For example, you can set url='jdbc:mysql://<mappingIP>:<mappingPort>/<databaseName>'. To obtain the values of the **Mapping IP Address** and **Mapping Port** parameters, perform the following steps:

1. Log on to the Realtime Compute console.
2. Move the pointer over the username in the upper-right corner.
3. In the list that appears, click **Project Management**.
4. In the left-side navigation pane, click **VPC Access Authorization**.
5. On the **VPC Access Authorization** page, view the values of the **Mapping IP Address** and **Mapping Port** parameters.



4.3. Overview of data storage

4.3.1. Overview

To facilitate data storage management, you can register data storage resources on the Realtime Compute development platform. This enables you to use the advantages of the one-stop Realtime Compute service. In Realtime Compute, you can manage multiple data storage systems, such ApsaraDB RDS, AnalyticDB for MySQL, and Tablestore. This one-stop management service allows you use manage data stores in the cloud on the Realtime Compute development platform without the need to navigate across multiple management consoles of different storage systems.

4.3.2. Types

Realtime Compute supports both streaming data storage and static data storage.

Streaming data storage

Streaming data storage systems provide inputs and outputs for downstream Realtime Compute jobs.

Streaming data storage

Storage system	Input	Output
DataHub	Supported	Supported
Log Service	Supported	Supported
MQ	Supported	Supported

Static data storage

Static data storage systems provide outputs for downstream Realtime Compute jobs and allow you to perform association queries.

Static data storage

Support	Dimension table	Output
ApsaraDB RDS	Supported	Supported
Tablestore	Supported	Supported

4.3.3. Registration and usage

This topic describes how to register and use external data stores in Realtime Compute.

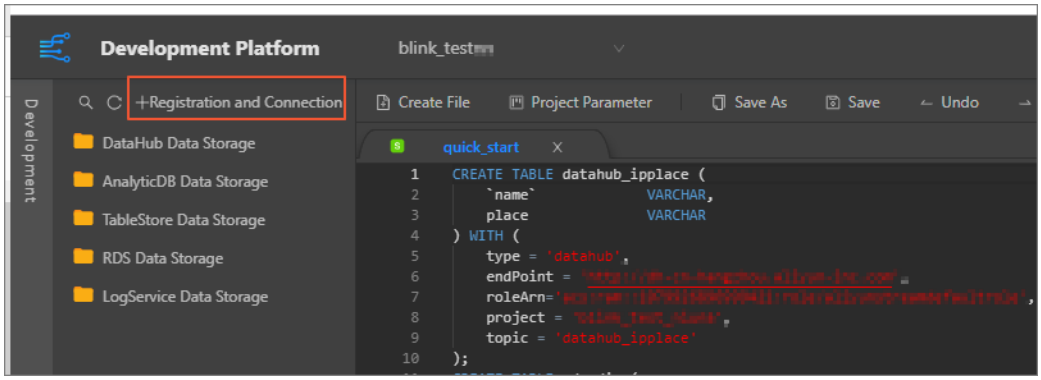
Note If a job requires the use of the data stores owned by another Apsara Stack tenant account, you can write DDL statements to reference the data stores. In the DDL statements, you must specify the AccessKey ID and AccessKey secret of the account. In this scenario, you cannot use the codeless UI to manage the data stores.

Register a data store

To register a data store, follow these steps:

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane of the page that appears, click the **Storage** tab. Select the folder for the data store that you want to register. Then, click **+Registration and Connection**.

   Register a data store



4. In the dialog box that appears, configure the required parameters and click **OK**.

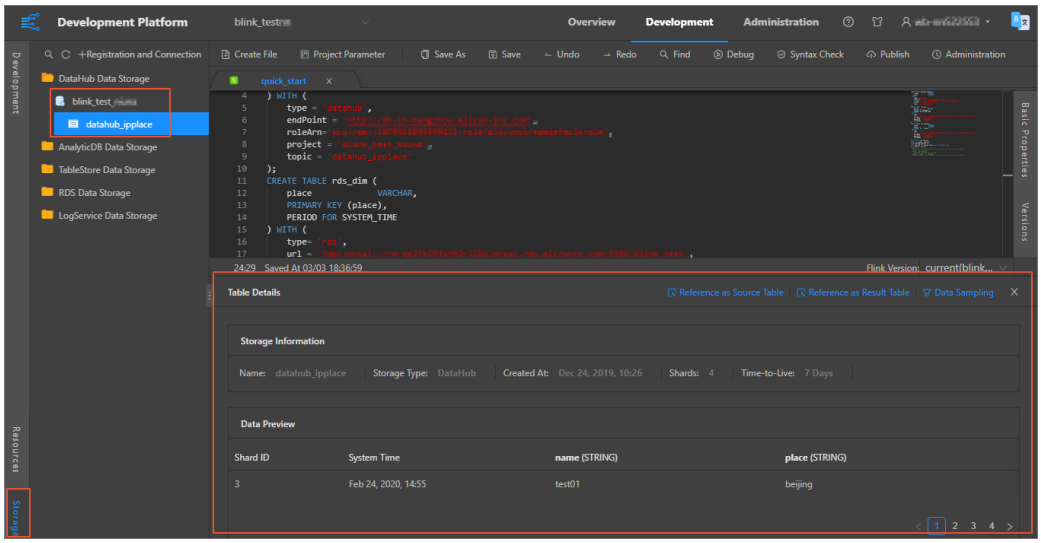
> **Note** After you enable the data store registration feature, you can only register data stores that are owned by your organization.

Preview data from a data store

Realtime Compute provides the data preview feature for each registered data store. To preview data, click the **Storage** tab and double-click the folder of the target data store in the left-side navigation pane. The following example shows how to preview data from a DataHub data store.

1. Log on to the Realtime Compute console. In the top-navigation bar, click Development. On the page that appears, click the **Storage** tab and double-click the **DataHub Data Storage** folder.
2. Double-click the target project and then the target topic to view the details.

   Table details



Use the auto DDL generation feature

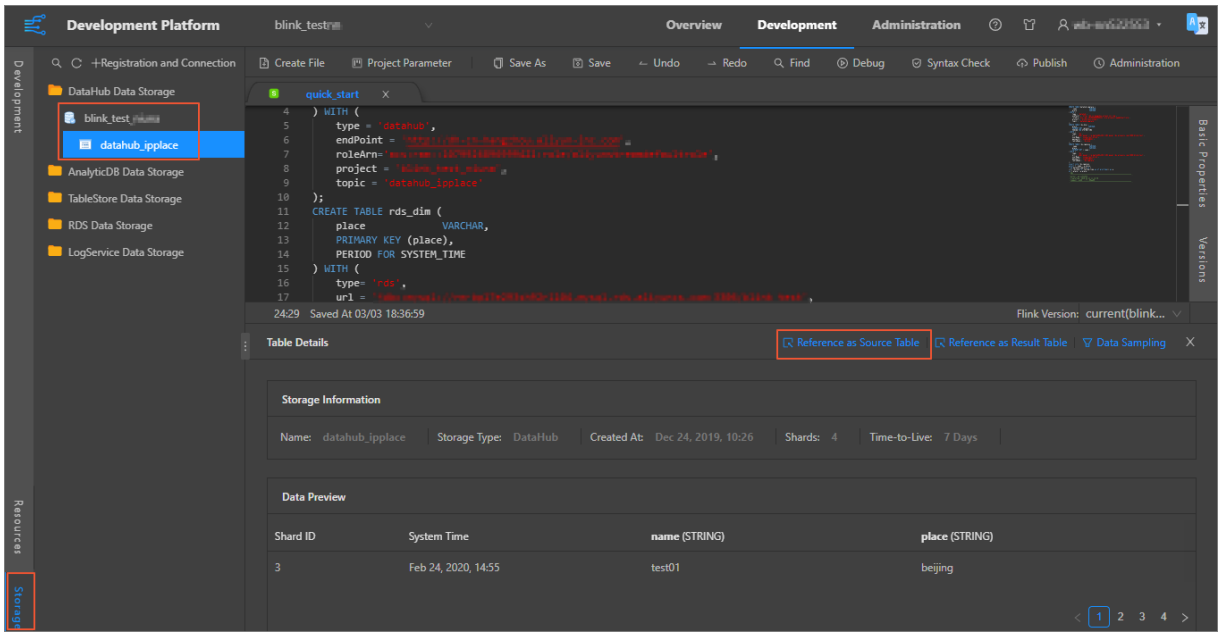
You must declare tables from external data stores before you can reference these tables. The following example shows how to reference a source table that contains streaming data:

```
CREATE TABLE in_stream( a varchar, b varchar, c timeStamp) with ( type='datahub', endPoint=
'http://dh-cn-hangzhou.aliyuncs.com', project='blink_test', topic='ip_count02', accessId='L
TAIYtaf******', accessKey='gUqyVwfkK2vfJI7jF90******');
```

The field names in the table that is referenced on the Development page must be the same as those in the DataHub topic. You must declare the field data types in the code based on the field type mapping between DataHub and Realtime Compute to ensure that Realtime Compute can identify the data. Realtime Compute offers the auto DDL generation feature. The following section describes how to use this feature.

1. In the left-side navigation pane of the **Development** page, click the **Storage** tab.
2. On the **Storage** tab, navigate through cascaded folders and nodes to find the target table. Then, double-click the name of the target table.
3. In the **Table Details** pane that appears, click **Reference as Source Table**, **Reference as Result Table**, or **Reference as Dimension Table** as required. Then, you can obtain the DDL statements that are automatically generated to reference the target table.

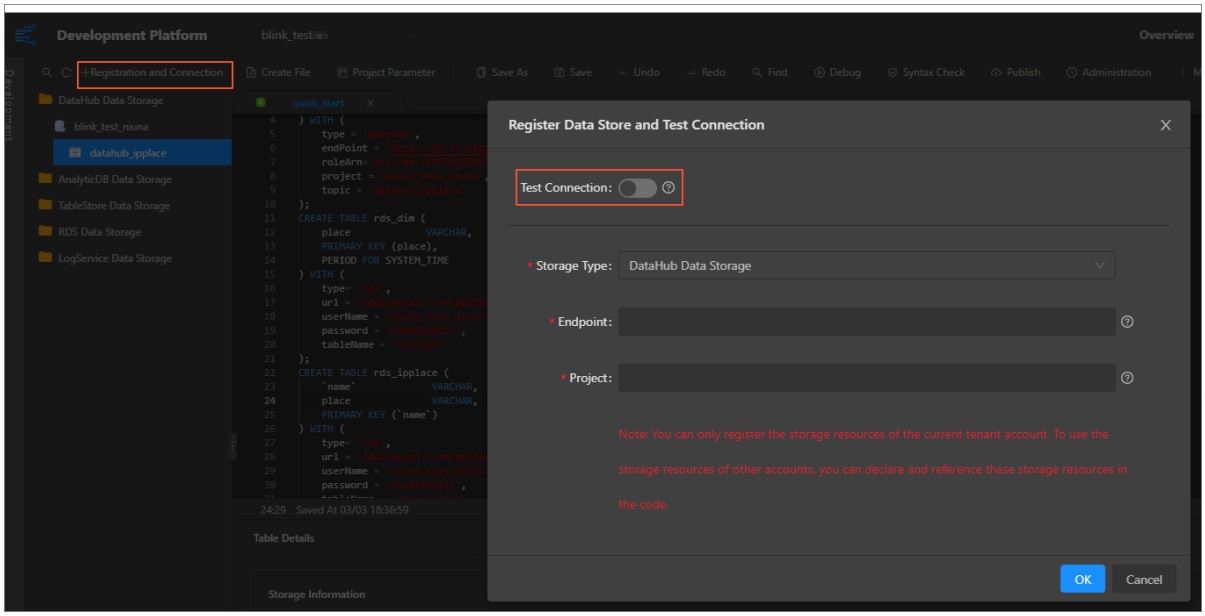
To reference a source table, log on to the Realtime Compute console and open the target SQL file on the Development page. Click the Storage tab, select the table for reference, and then click **Reference as Source Table**. The required DDL statements are displayed on the current page.



Test network connection

Realtime Compute offers the network connection test feature for data stores. This feature allows you to test the connection between Realtime Compute and a target data store. To enable the network connection test feature, follow these steps:

1. In the left-side navigation pane of the **Development** page, click the **Storage** tab.
2. In the upper-right corner of the **Storage** tab, click **+Registration and Connection**.
3. In the **Register Data Store and Test Connection** dialog box, turn on **Test Connection**.



Example: Reference data stores owned by another level-1 organization

You can only register and use data stores that are owned by your level-1 organization. To use data stores that are owned by another level-1 organization, write DDL statements to create a reference to these data stores. For example, if a user from Organization A wants to use the data stores owned by Organization B, the user can enter the following DDL statements:

```
CREATE TABLE in_stream( a varchar, b varchar, c timeStamp) with ( type='datahub', endPoint='http://dh-cn-hangzhou.aliyuncs.com', project='blink_test', topic='ip_count02', accessId='AccessKey ID authorized by Organization B users', accessKey='AccessKey secret authorized by Organization B users');
```

4.4. Register a DataHub data store

DataHub, an Alibaba Cloud streaming data service, is a real-time data distribution platform designed to process streaming data. You can publish and subscribe to applications for streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data. Realtime Compute often uses DataHub to store source and result tables that contain streaming data.

Register a DataHub project

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Storage** tab.
4. Right-click the **DataHub Data Storage** folder and select **Register Data Store** to register a DataHub project in Realtime Compute. Parameter description

Parameter	Description

Parameter	Description
Test Connection	Specifies whether to enable the network connection test feature. Network connection tests are automatically performed on data stores that can be registered in Realtime Compute. To test the connection between Realtime Compute and data stores that cannot be registered, turn on the Test Connection switch.
Storage Type	The type of the data store. DataHub Data Storage is selected by default.
Endpoint	The endpoint of DataHub. The endpoint of DataHub varies with regions. For more information about endpoints, contact your administrator. **Note** To specify this parameter for Apsara Stack, contact your Apsara Stack administrator to obtain the endpoint of DataHub.
Project	The name of the DataHub project. **Note** You can only register DataHub projects that are owned by your level-1 organization. For example, if DataHub Project A is owned by Organization A, users from Organization B cannot register Project A in Realtime Compute.
AccessKey ID	The AccessKey ID of the current account.
AccessKey Secret	The AccessKey secret of the current account. The AccessKey secret enables Realtime Compute to access the DataHub project.

Scenarios

DataHub is a streaming data storage system that can be used to store source and result tables. However, it cannot be used to store dimension tables for Realtime Compute.

FAQ

Q: Why am I unable to register a DataHub project in Realtime Compute?

A: Realtime Compute uses a storage software development kit (SDK) to access different data stores. The Storage tab in the Realtime Compute console only helps you manage data from different data stores. You can perform the following operations to troubleshoot registration errors:

- Check whether you have created a DataHub project and have the permissions to access the project. You can log on to the DataHub console and check whether you can access the project.
- Check whether you are the project owner. You can only register DataHub projects that are owned by your level-1 organization. For example, if DataHub Project A is owned by Organization A, users from Organization B cannot register Project A in Realtime Compute.
- Check whether you have specified the correct DataHub endpoint and project name.
- Check whether you have specified a classic network endpoint for the Endpoint parameter. If you specify a VPC endpoint, the DataHub project will fail to be registered.
- Check whether you have registered the DataHub project. Realtime Compute provides a registration

check mechanism to prevent duplicate registration.

Q: Why does Realtime Compute only support time-based sampling?

A: DataHub stores streaming data, and you can only specify time parameters in the API. Therefore, Realtime Compute supports only time-based sampling.

4.5. Register a Log Service data store

Log Service (formerly known as SLS) provides an end-to-end solution for log management. You can use Log Service to collect, subscribe to, dump, and query large amounts of log data. If you use Log Service to manage Elastic Compute Service (ECS) logs, you can use Realtime Compute together with Log Service to process ECS logs. You do not need to transfer data between these systems.

Register a Log Service project

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Storage** tab.
4. Right-click the **LogService Data Storage** folder and select **Register Data Store** to register a Log Service project in Realtime Compute. Parameter description

Parameter	Description
Test Connection	Specifies whether to enable the network connection test feature. Network connection tests are automatically performed on data stores that can be registered in Realtime Compute. To test the connection between Realtime Compute and data stores that cannot be registered, turn on Test Connection.
Storage Type	The type of the data store. By default, LogService Data Storage is selected.
Endpoint	The endpoint of Log Service. The endpoint of Log Service varies based on the regions. **Note** To obtain the endpoint of Log Service, contact the Apsara Stack system administrator.
Project	The name of the Log Service project. **Note** You can register only Log Service projects that are owned by your level-1 organization. For example, if Log Service Project A is owned by Organization A, users from Organization B cannot register Project A in Realtime Compute.
AccessKey ID	The AccessKey ID of the current Alibaba Cloud account.
AccessKey Secret	The AccessKey secret of the current Alibaba Cloud account. This information allows Realtime Compute to access the Log Service project.

Scenarios

Log Service is a streaming data storage system that can be used to store source tables and result tables. However, Log Service cannot be used to store dimension tables for Realtime Compute.

FAQ

- Q: Why am I unable to register a Log Service project in Realtime Compute?

  A: Realtime Compute uses a storage SDK to access different data stores. The Storage tab in the Realtime Compute console only helps you manage data from different data stores. You can perform the following operations to troubleshoot registration errors:

  - Check whether a Log Service project is created and whether you have the permissions to access the project. Log on to the Log Service console and check whether you can access the project.
  - Check whether you are the owner of the Log Service project. You can register only Log Service projects that are owned by your level-1 organization. For example, if Log Service Project A is owned by Organization A, a user from Organization B cannot register Project A in Realtime Compute.
  - Check whether the Log Service endpoint and the project name that you entered are correct.

    > **Note** The endpoint must start with http and cannot end with a forward slash (/). For example, `http://cn-hangzhou.log.aliyuncs.com` is correct, and `http://cn-hangzhou.log.aliyuncs.com/` is incorrect.

  - Check whether the Log Service project is registered. Realtime Compute provides a registration check mechanism to prevent duplicate registration.

- Q: Why is only time-based sampling supported?

  A: Log Service stores streaming data, and you can specify only time parameters for Log Service in the APIs. Therefore, Realtime Compute supports only time-based sampling.

  > **Note** If you want to use the search feature of Log Service, you must log on to the Log Service console.

4.6. Register a Tablestore data store

Tablestore is a NoSQL database service that is based on the Apsara distributed system. Tablestore allows you to store and access large amounts of structured data in real time. Tablestore features massive data storage and low access delays, which makes it suitable to store dimension tables and result tables for Realtime Compute.

Register a Tablestore instance

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Storage** tab.
4. Right-click **TableStore Data Storage** and select **Register Data Store**. In the dialog box that appears, register a Tablestore instance in Realtime Compute. Parameters

Parameter	Description
Test Connection	Specifies whether to enable the network connection test feature. Network connection tests are automatically performed on data stores that can be registered in Realtime Compute. To test the connection between Realtime Compute and data stores that cannot be registered, turn on the **Test Connection** switch.
Storage Type	The type of the data store. **TableStore Data Storage** is selected by default.
Endpoint	The endpoint of the Tablestore instance. You must enter the internal endpoint of the Tablestore instance. You can log on to the Tablestore console to view the internal endpoint of the instance.
Instance Name	The name of the Tablestore instance.
AccessKey ID	The AccessKey ID of the current account.
AccessKey Secret	The AccessKey secret of the current account. The AccessKey secret enables Realtime Compute to access the Tablestore instance.

4.7. Register an ApsaraDB RDS data store

This topic describes how to register and use an ApsaraDB RDS data store in Realtime Compute.

Introduction to ApsaraDB RDS

ApsaraDB RDS offers a stable, reliable, and scalable online database service. Based on the Apsara distributed operating system and high performance SSD storage, ApsaraDB RDS supports a wide range of engines, such as MySQL, PostgreSQL, and Postgres Plus Advanced Server (PPAS, highly compatible with Oracle). Realtime Compute supports the following ApsaraDB RDS engines: MySQL and PostgreSQL.

The performance of Tablestore in high concurrency scenarios where large amounts of data need to be processed is higher than that of ApsaraDB RDS. The performance of ApsaraDB RDS is restricted by the limits of relational models. Therefore, ApsaraDB RDS is often used to store result tables for Realtime Compute. In low concurrency scenarios where a small number of data needs to be processed, ApsaraDB RDS can be used to store dimension tables.

> **Note** Realtime Compute uses relational databases, such as ApsaraDB RDS for MySQL, to store result data. Distributed Relational Database Service (DRDS) and ApsaraDB RDS connectors are used. If Realtime Compute frequently writes data to a DRDS table or an ApsaraDB RDS table, deadlocks may occur. In scenarios that require high queries per second (QPS), high transactions per second (TPS), or highly concurrent write operations, we recommend that you do not use DRDS or ApsaraDB RDS to store the result tables of Blink jobs. To prevent deadlocks, we recommend that you use Tablestore to store result tables.

Register an ApsaraDB RDS instance

1. Log on to the Realtime Compute console.

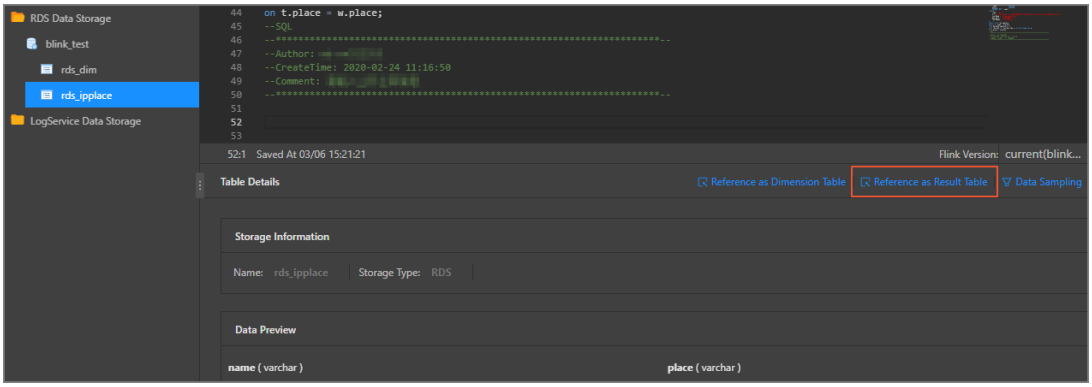
2. In the top navigation bar, click **Development**.
3. In the left-side navigation pane, click the **Storage** tab.
4. Right-click **RDS Data Storage**, and select **Register Data Store**. In the dialog box that appears, register an ApsaraDB RDS instance in Realtime Compute. Parameter description

Parameter	Description
Test Connection	Specifies whether to enable the network connection test feature. Network connection tests are automatically performed on data stores that can be registered in Realtime Compute. To test the connection between Realtime Compute and data stores that cannot be registered, turn on Test Connection.
Storage Type	The type of the data store. RDS Data Storage is selected by default.
URL	The URL that is used to access the ApsaraDB RDS database.
DBName	The name of the ApsaraDB RDS database to be accessed by Realtime Compute. **Note** This parameter specifies the ApsaraDB RDS database name instead of the ApsaraDB RDS instance name. ApsaraDB RDS uses whitelists for access control to ensure system security. The IP addresses of the Realtime Compute console and worker nodes must be added to the whitelists of ApsaraDB RDS. Otherwise, Realtime Compute may fail to connect to ApsaraDB RDS. For more information, see Specify whitelist settings.
User Name	The username that is used to log on to the ApsaraDB RDS database.
Password	The password that is used to log on to the ApsaraDB RDS database.
Engine	The type of the ApsaraDB RDS database. Valid values: ○ **mysql** ○ **postgresql** ○ **sqlserver**

Reference an ApsaraDB RDS table as a result table

After you register an ApsaraDB RDS data store, double-click the ApsaraDB RDS database, double-click the ApsaraDB RDS table that you want to reference as a result table, and then click **Reference as Result Table**.

Reference an ApsaraDB RDS table as a result table



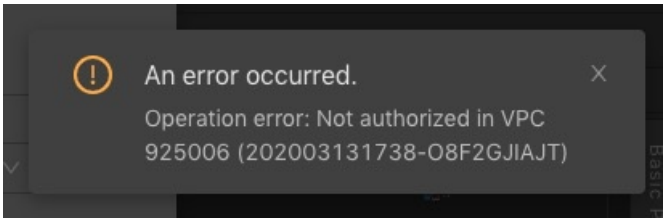
After you click Reference as Result Table, Realtime Compute automatically generates the related DDL statements on the current page.

Result

```
    place              VARCHAR,
    PRIMARY KEY (`name`)
) WITH (
    type= 'rds',
    url = '',
    userName = '',
    password = '',
    tableName = ''
);
```

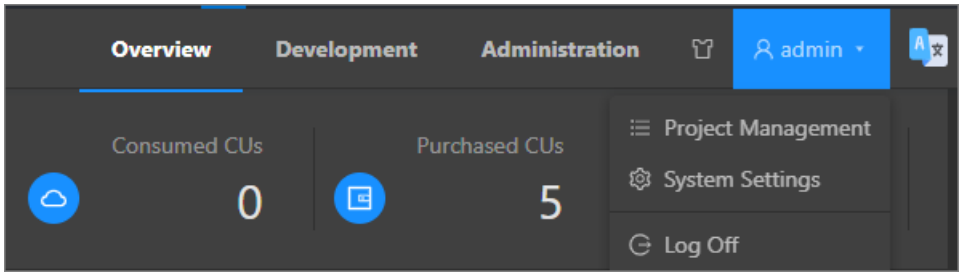
If the following error message appears, troubleshoot and rectify the fault in the following way.

Error message



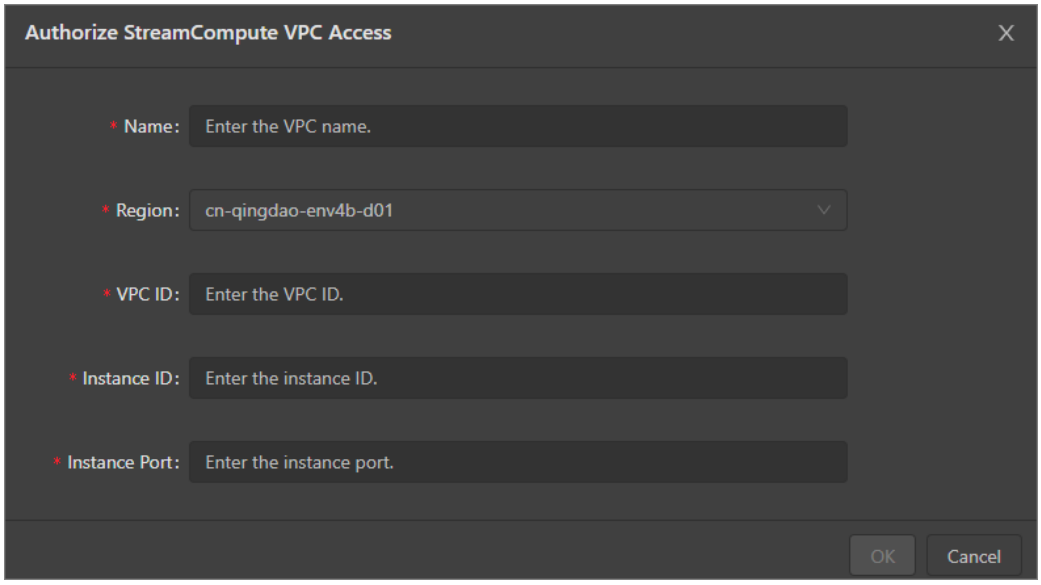
The error occurs because a VPC instead of a classic network was selected when you created the ApsaraDB RDS instance. You can perform the following steps to rectify this fault:

1. Move the pointer over the administrator icon, as shown in the following figure.

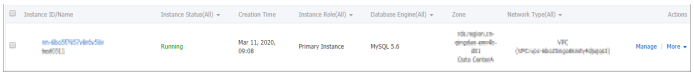


2. Click **System Settings**.
3. In the left-side navigation pane, click **VPC Access Authorization**.
4. Click **Add Authorization**. The **Authorize StreamCompute VPC Access** page appears.

Authorization



Parameter description

Parameter	Description
Name	The name of the VPC.
Region	The region where the ApsaraDB RDS instance resides.
VPC ID	The ID of the VPC.
Instance ID	The ID of the ApsaraDB RDS instance. You can log on to the ApsaraDB RDS console and view the instance ID. Instance information 
Instance Port	The port that is used to access the ApsaraDB RDS instance. To view the internal port number, log on to the ApsaraDB RDS console, click the ID or name of the instance that you want to access in the Instance ID/Name column. On the page that appears, view the internal port number in the Basic Information section.

5. Register the ApsaraDB RDS instance. You must specify the required parameters during the registration.

You can register only data storage resources that are owned by your level-1 organization. For example, if ApsaraDB RDS Instance A is owned by Organization A, a user from Organization B cannot register ApsaraDB RDS Instance A in the Realtime Compute console. To use Instance A in a stream processing job, the user from Organization B must use SQL code to create a reference to Instance A.

> **Note** If you want to use the ApsaraDB RDS storage resources owned by your level-1 organization, we recommend that you do not use SQL code to create a reference to these resources.

The user from Organization B must also specify the following parameters in the WITH clause based on the information of Instance A: url, userName, password, and tableName.

Configuration category

```
CREATE TABLE datahub_output (
    id            varchar
) WITH (
    type= 'rds',
    url = '                ',
    userName = '                ',
    password = '                ',
    tableName = 'datahub_output'
);
```

To use ApsaraDB RDS storage resources by writing SQL code, the user from Organization B must specify whitelist settings.

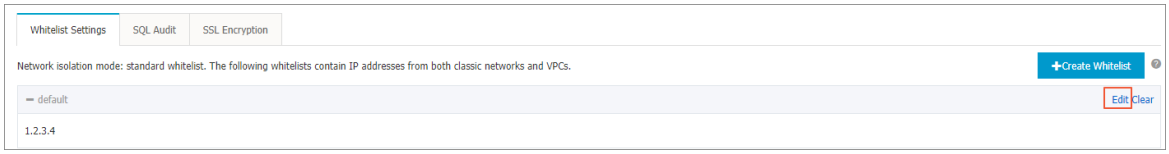
Specify whitelist settings

Some data stores use whitelists for access control to ensure high-level security. These data stores allow access only from the IP addresses that are added to the whitelists. This prevents unauthorized Apsara Stack services from accessing data in these data stores. For example, a newly created ApsaraDB RDS database denies all access. You must add IP addresses to a whitelist of the ApsaraDB RDS database to allow access to the database.

ApsaraDB RDS can be accessed from both external and internal networks. To enable Realtime Compute to access ApsaraDB RDS, you must add the CIDR blocks of Realtime Compute to a whitelist of the ApsaraDB RDS database.

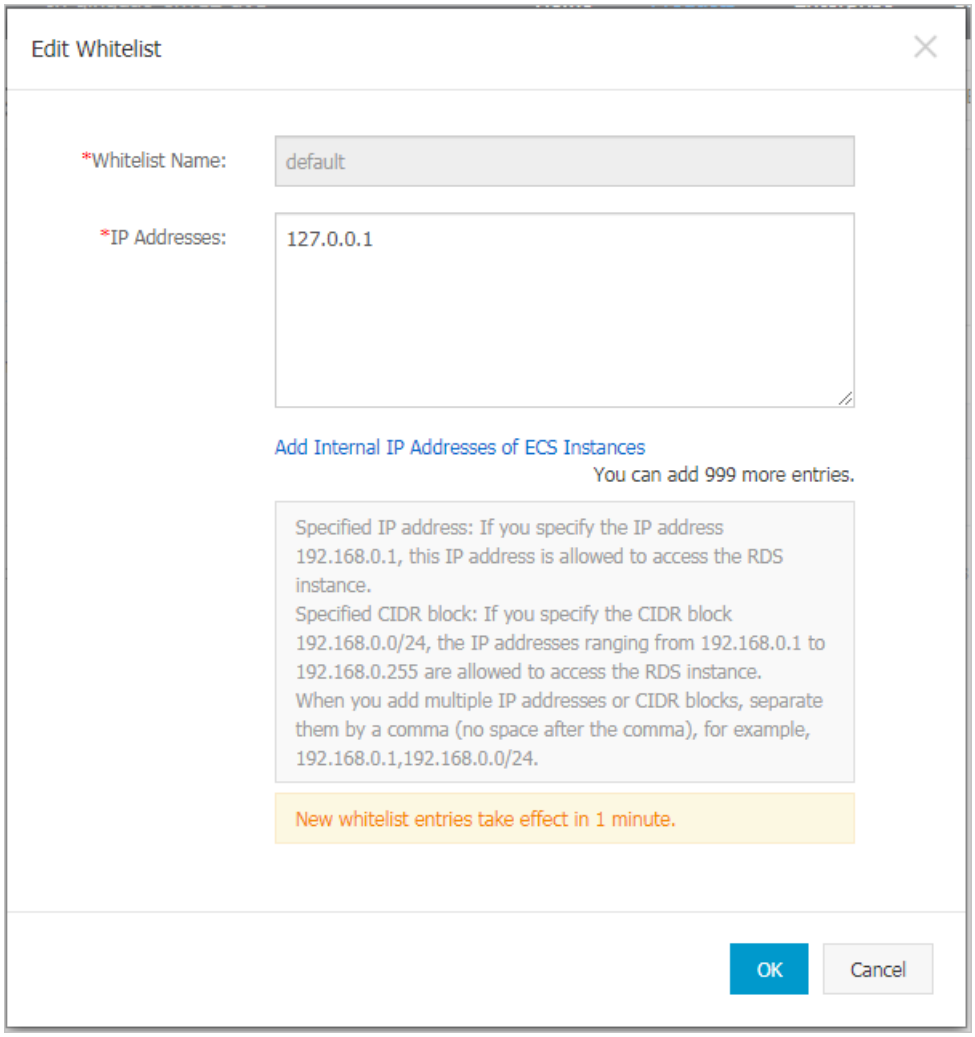
Procedure:

1. Log on to the ApsaraDB RDS console.
2. On the **Instances** page, click the ID of an instance in the **Instance ID/Name** column.
3. In the left-side navigation pane, click **Data Security**.
4. On the **Whitelist Settings** tab, click **Edit** that corresponds to the **default** whitelist.



> **Note**
>
> - If you want to connect an ECS instance to an ApsaraDB RDS instance by using an internal endpoint, you must make sure that the two instances are in the same region and have the same network type. Otherwise, the connection fails.
> - You can also click **Create Whitelist** to create a whitelist.

5. In the **Edit Whitelist** dialog box, specify the IP addresses or CIDR blocks that are used to access the instance, and then click **OK**.



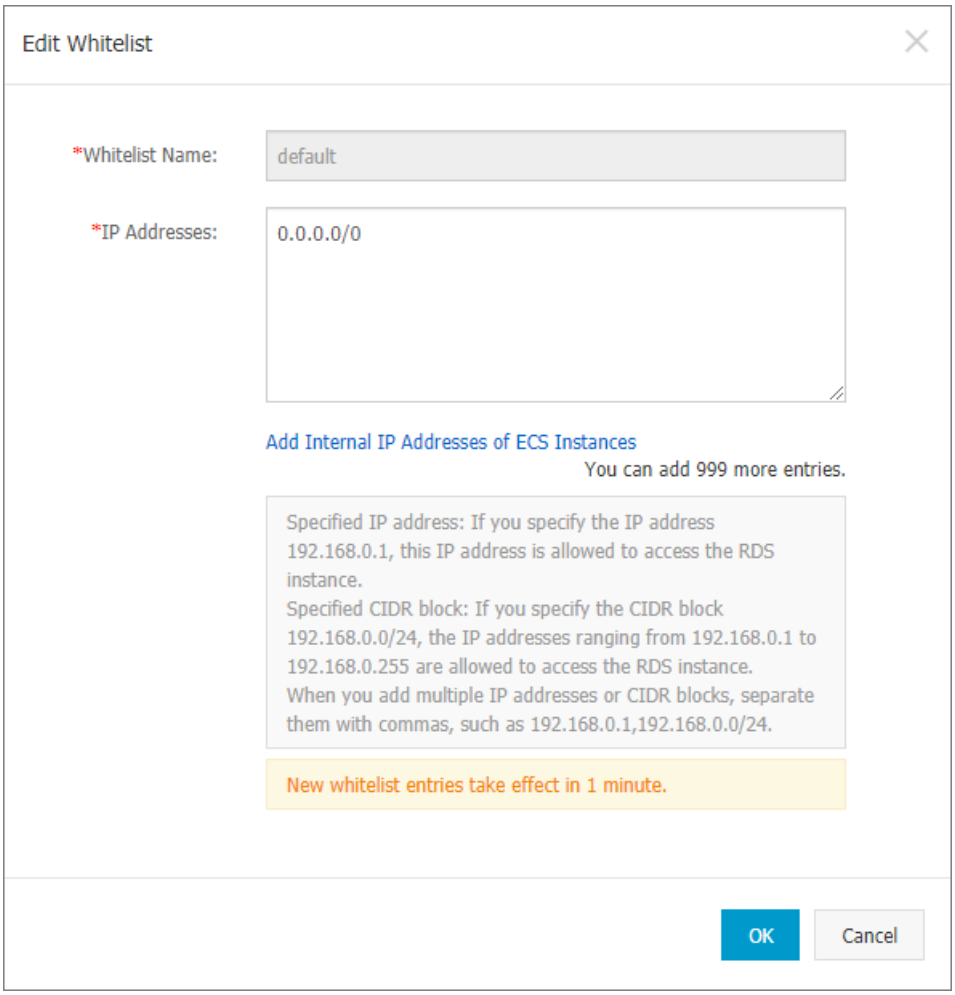
- If you specify the 10.10.10.0/24 CIDR block, IP addresses in the 10.10.10.X format are allowed to access the ApsaraDB RDS instance.
- If you want to add multiple IP addresses or CIDR blocks, separate entries with commas (without spaces), such as 192.168.0.1,172.16.213.9.
- After you click **Add Internal IP Addresses of ECS Instances**, the IP addresses of all the ECS instances under your Apsara Stack account are displayed. You can select the required IP addresses to add to the whitelist.

> **Note** If you add a new IP address or CIDR block to the **default** whitelist, the default address 127.0.0.1 is automatically deleted.

FAQ

- Fault description

A stack exception occurs while the system is running, as shown in the following figure.



- Solution

  Add the IP address of your region to an RDS whitelist. For more information, see Specify whitelist settings.

5.Data Development

5.1. Create a job

This topic describes how to create a Realtime Compute job.

Procedure

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Development**.
3. In the top navigation bar, click **Create File**.
4. In the **Create File** dialog box, configure the parameters. The following table describes the parameters.

Parameter	Description
File Name	The name of the file. The name must be 3 to 64 characters in length and can contain only lowercase letters, digits, and underscores (_). The name must start with a lowercase letter.
File Type	The type of the file. Valid values: FLINK_STREAM/SQL and FLINK_STREAM/DATASTREAM.
Storage Path	The folder in which the job SQL file is located. You can click the folder icon on the right side of an existing folder and create a subfolder.

5. Click **OK**.

5.2. Development

5.2.1. SQL code assistance

The development platform of Realtime Compute offers a complete set of SQL tools in the integrated development environment (IDE). These tools provide the following features to help you with Flink SQL-based development:

- Syntax check

  On the **Development** page of Realtime Compute, the revised script is automatically saved. When the script is saved, an SQL syntax check is automatically performed. If a syntax error is detected, the **Development** page shows the row and column where the error is located, and the cause of the error.

- Intelligent code completion

  When you enter SQL statements on the **Development** page of Realtime Compute, auto completion popups about keywords, built-in functions, tables, or fields are automatically displayed.

- Syntax highlighting

  Flink SQL keywords are highlighted in different colors to differentiate data structures.

5.2.2. SQL code version management

Realtime Compute provides key features that help you complete development tasks, such as coding assistance and code version management. A new code version is generated each time you publish a job SQL file. The code version management feature allows you to track code changes and roll back to an earlier version if required.

- Manage code versions

  On the **Development** page, you can manage Flink SQL code versions. A new code version is generated each time you publish a job SQL file. You can use the code version management feature to track versions, modify the code, and roll the code back to an earlier version.

  On the **Versions** tab on the right side of the **Development** page, click **More** in the **Actions** column to manage code versions.

  - **Compare**: Check the differences between the current version and an earlier version.
  - **Rollback**: Roll back to an earlier version.
  - **Delete**: Delete an earlier version.
  - **Locked**: Lock the current version.

    > **Note** You cannot submit a new version before you unlock the SQL file.

- Delete code versions

  A snapshot of a code version is created each time you submit an SQL file for publishing a job. This allows you to track code changes. The maximum number of code versions has been specified. If you use Apsara Stack, a maximum of 20 code versions can be published. To find out the maximum number of code versions in other environments, contact the system administrator. If the number of code versions reaches the upper limit, an error message is displayed to alert you to delete one or more earlier versions.

  In this scenario, you must delete one or more earlier versions before you publish new versions. To do this, click the **Versions** tab on the right side of the **Development** page, click **More** in the **Actions** column, and select **Delete** to delete expired versions that are no longer needed.

5.2.3. Data store management

The Development page of the Realtime Compute console provides an easy and effective method to manage data stores. For example, you can register external data stores to reference the data stores.

- Data preview

  The Development page of the Realtime Compute console allows you to preview data of multiple types of data stores. Data preview allows you to analyze the characteristics of upstream and downstream data, identify key business logic, and complete development tasks with high efficiency.

- Auto DDL generation

  Realtime Compute can automatically generate DDL statements to reference external data stores. This feature provides a simple method to write SQL statements for stream processing jobs. This improves overall efficiency and reduces errors when you write SQL statements.

5.3. Debug job code

The Realtime Compute development platform provides a simulated running environment where you can customize uploaded data, simulate operations, and check outputs.

After you write SQL code that implements the computing logic, perform the following steps to debug the code:

1. Log on to the Realtime Compute console to go to the homepage of Alibaba Cloud Realtime Compute.
2. In the top navigation bar, click Development.
3. In the left-side navigation pane, click **Development**.
4. On the **Development** tab, double-click the folder and file name to open the job file.
5. In the top menu bar, click **Syntax Check**.

   > **Note** You can use the syntax check feature to check whether the SQL file includes syntax errors. Error messages are displayed for syntax errors.

6. In the top menu bar, click **Debug**. On the **Debug File** page, debug your SQL code.

   The test data for debugging can be acquired by using either of the following two methods:

   - Upload local data.
     a. Click **Download Template**.
     b. Prepare test data based on the template.
     c. Click **Upload**. After the file is uploaded, you can view the uploaded data in the data preview section.
   - Sample online data.
     a. Click **Random Online Data Sampling** or **Sequential Online Data Sampling**.
     b. View the sampled data in the data preview section.
7. Click **OK**.
8. In the output window, view the debugging result.

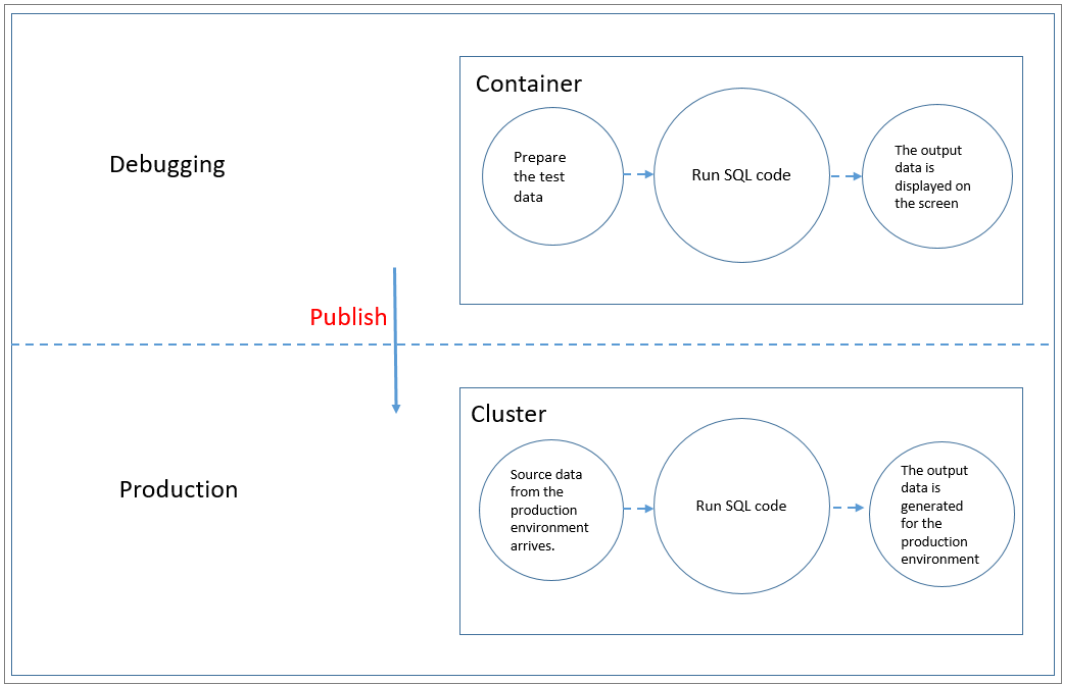
The debugging feature of Realtime Compute provides the following functions:

- Enables isolation between debugging and production environments.

  In the debugging environment, the Flink SQL code runs in a separate container, and computing result data is only displayed on the screen of the Development page. In this way, the debugging does not affect the running jobs and data stores in the production environment.

  In the debugging phase, result data is not written to external data stores. In the production environment, failures may occur due to format errors when result data is written to the target data stores. Such failures cannot be identified or prevented in the debugging phase, and can be detected only while jobs are running. For example, failures may occur in the production environment if your result data is too long. This occurs when the result data is written to a result table in ApsaraDB RDS and the length of character strings reaches the upper limit for an ApsaraDB RDS table. The Realtime Compute team is working on support for writing result data to external data stores in the production environment. This allows you to effectively simulate the production environment and resolve more issues in the debugging phase.

  Isolation between debugging and production environments



- Supports the customization of test data.

  In the debugging environment, Realtime Compute does not read data from source data stores, such as DataHub topics that store source tables and ApsaraDB RDS instances that store dimension tables. You must create a set of test data and upload the test data on the Development page.

  To make the debugging feature easy to use, Realtime Compute provides a template of test data for each type of job. You can download the template and enter your test data.

  > **Note** We recommend that you use the templates to prevent errors.

- Specifies a separator.

  A comma (,) is used as the separator in files for debugging by default. The following example shows a file for debugging:

  ```
  id,name,age
  1,alicloud,13
  2,stream,1
  ```

  If the separator is not specified, a comma (,) is used to separate fields. If you need to use a JSON string as the field data and the string contains commas (,), you must specify another character as the separator.

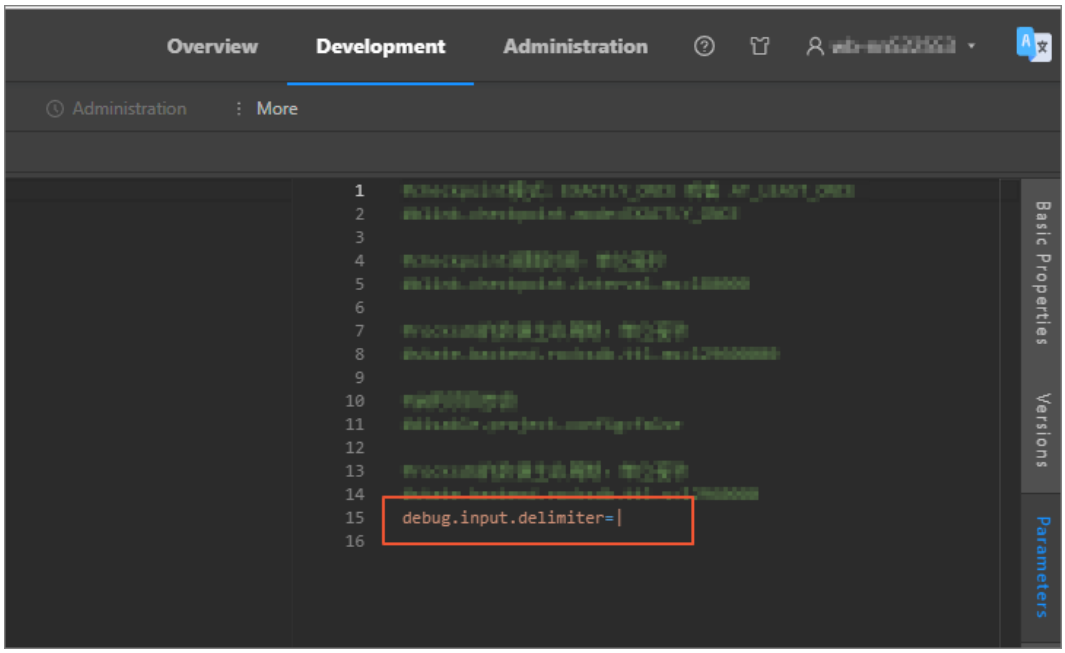
  > **Note** Realtime Compute allows you to specify a character as the separator, but not a multi-character string, such as aaa.

  ```
  id|name|age
  1|alicloud|13
  2|stream|1
  ```

  In this example, set `debug.input.delimiter=|` .

  Specify a separator

Specify a separator.



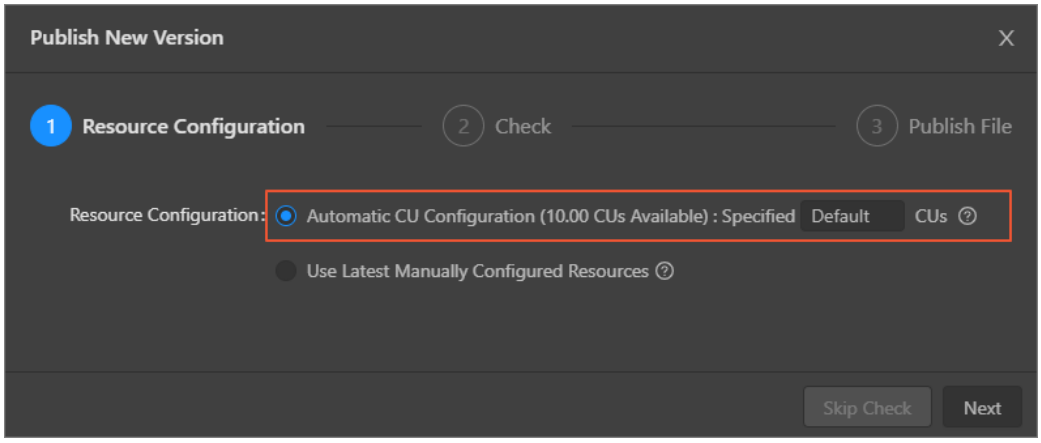
5.4. Publish a job SQL file

After you have created and debugged a job Flink SQL file, you can publish the SQL file and manage the job in the production environment.

Procedure

1. Log on to the Realtime Compute console.
2. In the top navigation bar, Click **Development**.
3. In the top menu bar, click **Publish**.
4. In the dialog box that appears, select **Automatic CU Configuration**. If you are performing automatic configuration for the first time, we recommend that you use the default number of CUs. Click **Next**.

   Configure resources



5. Check the data. After the check is completed, click **Next**.
6. Click **Publish**.

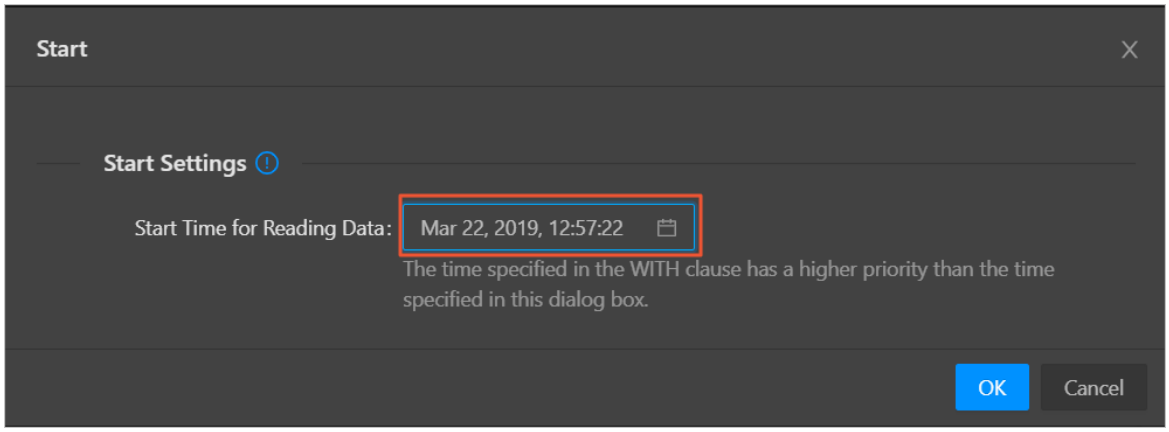
7. Go to the **Administration** page to start the job.
   i. In the top navigation bar, click **Administration**.
   ii. On the **Administration** page, find the target job, and click **Start** in the **Actions** column.

5.5. Start a job

After you develop and publish a job, you can start the job on the Administration page.

Procedure

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Administration**.
3. Find the job that you want to start, and click **Start** in the **Actions** column.
4. In the **Start** dialog box, set the **Start Time for Reading Data** parameter.



5. Click **OK**. The job is started.

   Start Time for Reading Data indicates the time at which the system starts to read data from the source table.

   - If you select the current time, Realtime Compute reads the data generated after the current time.
   - If you select a previous time, Realtime Compute reads the data generated after the specified time. This is used to track historical data.

5.6. Suspend a job

After you modify the resource configuration of a job, you must suspend and resume the job to make the changes take effect. This topic describes how to suspend a job.

Context

> **Notice**
>
> - You can only **suspend** a job that is in the **Running** state.
> - If you **suspend** a job, its task status is not cleared. For example, if the job you **suspend** is running a COUNT operation, the COUNT operation continues from the last successful checkpoint after you **resume** the job.
> - The Suspend (checkpoint) operation is supported in Realtime Compute V3.5.0 and later. If your Realtime Compute is earlier than V3.5.0, the following error message appears when you try to perform this operation: **An error occurred. System error: The Blink version is abnormal. Error reason: blink version >= blink-3.5 is required, instance blink-3.4.4.**

Procedure

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Administration**.
3. On the **Administration** page, find the job that you want to suspend, and click **Suspend** in the **Actions** column.

> **Note** The **Suspend (checkpoint)** operation in **More** suspends the job and triggers a checkpoint event. Therefore, the time consumed to suspend a job by performing the **Suspend (checkpoint)** operation is longer than that by performing the **Suspend** operation.

5.7. Terminate a job

After you modify the SQL logic, change the job version, add parameters to the WITH clause, or add job parameters for a job, you must terminate and then start the job to make the changes take effect. This topic describes how to terminate a job.

> **Notice**
>
> - You can only **terminate** a job that is in the **Running** or **Starting** state.
> - If you **terminate** a job, its task status is cleared. For example, if the job you **terminate** is running a COUNT operation, the COUNT operation starts from 0 after you **start** the job.
> - The **Terminate (checkpoint)** operation is supported in Realtime Compute V3.5.0 and later. If your Realtime Compute version is earlier than V3.5.0, the following error message appears when you try to perform this operation: **An error occurred. System error: The Blink version is abnormal. Error reason: blink version >= blink-3.5 is required, instance blink-3.4.4.**

To terminate a job, perform the following steps:

1. Log on to the Realtime Compute console.
2. In the top navigation bar, click **Administration**.
3. On the **Administration** page, find the job that you want to terminate, and click **Terminate** in the **Actions** column.

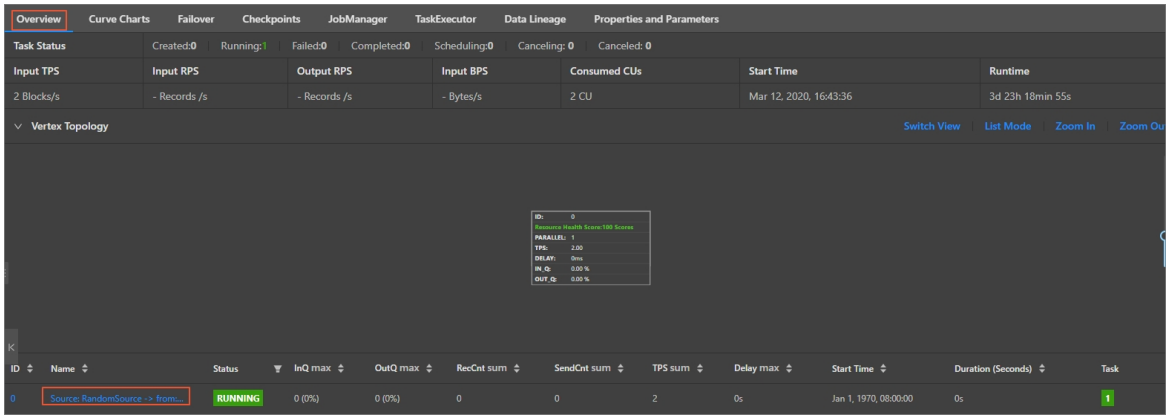
> **Note** The **Terminate (checkpoint)** operation under **More** is different from the **Terminate** operation. The system triggers a checkpoint when you perform the **Terminate (checkpoint)** operation to terminate a job. Therefore, the time consumed to terminate a job by performing the **Terminate (checkpoint)** operation is longer than that by performing the **Terminate** operation. The job status is cleared after the job is terminated. The Terminate (checkpoint) operation has other functions in some scenarios. For example, if the upstream storage system is Message Queue for Apache Kafka, the system submits an offset each time it triggers a checkpoint. This ensures that the number of offsets submitted to the Kafka server is consistent with the amount of data consumed.

5.8. View logs

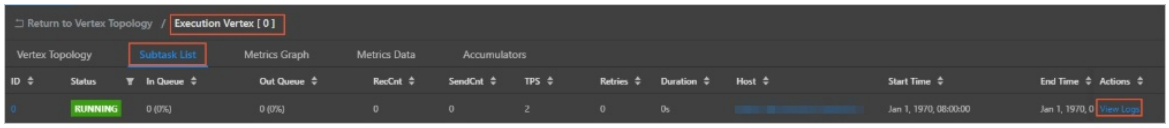
You can view the operational logs of a job to learn the job operation information. This topic describes how to view job logs.

Procedure

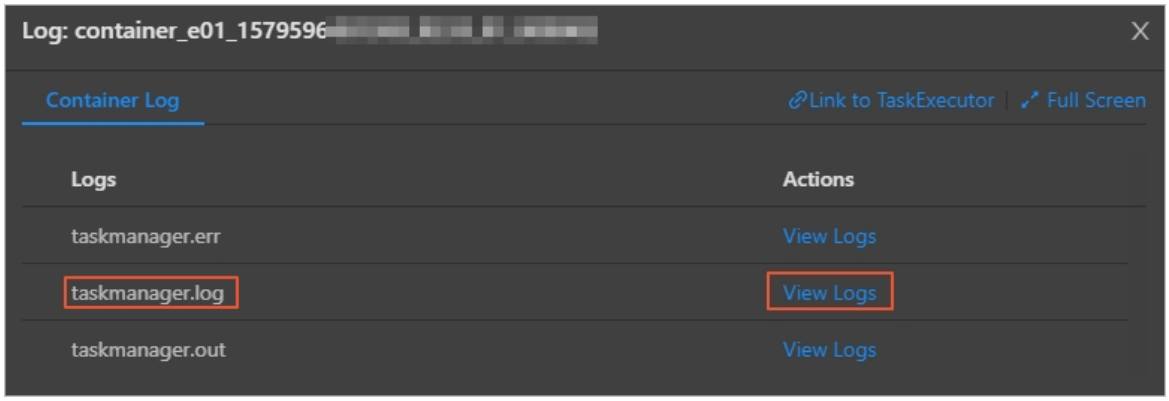
1. Go to the **Administration** page in Realtime Compute.
   i. Log on to the Realtime Compute console.
   ii. In the top navigation bar, click **Administration**.
   iii. On the **Jobs** page, click the name of the job whose logs you want to view in the **Job Name** column.
2. At the bottom of the **Overview** tab, click the name of the desired vertex.



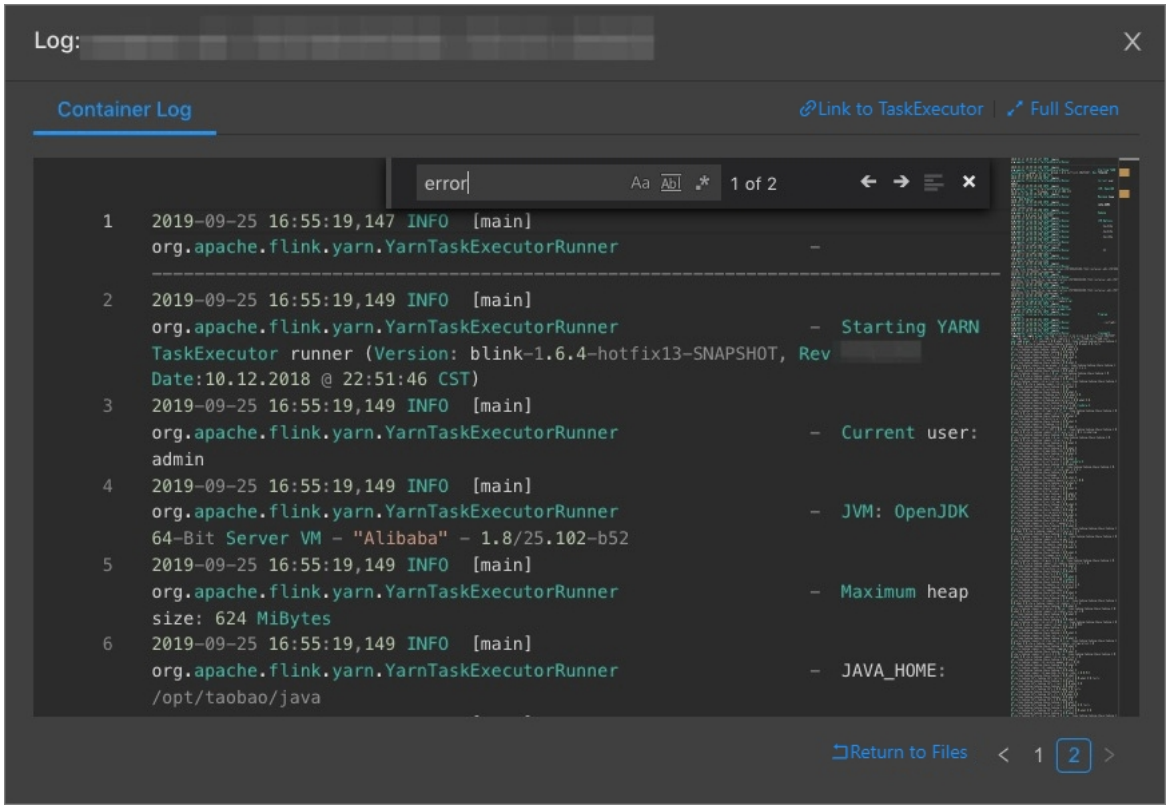
3. On the **Execution Vertex** page, click the **Subtask List** tab. Then, find the desired subtask and click **View Logs** in the **Actions** column.



4. In the **Log** dialog box, click **View Logs** for **taskmanager.log** in the **Actions** column.



5. On the **Container Log** tab, view the log entries.



> **Note** You can press **Ctrl+F** for Windows or **cmd+F** for MacOS to search for specified log entries. We recommend that you view the log entries from the last page. The first error recorded in the log describes the Root cause of the job error.