

ALIBABA CLOUD

Alibaba Cloud

**MaxCompute
Tutorials**

Document Version: 20200927

 **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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. Build an online operation analysis platform	05
1.1. Business scenarios and development process	05
1.2. Prepare the environment	05
1.3. Prepare data	12
1.4. Build data models and analyze data	18
1.4.1. Create tables	18
1.4.2. Design workflows	26
1.4.3. Configure nodes	27
1.4.4. Commit and test nodes	30
1.5. Display data on dashboards	34

1. Build an online operation analysis platform


1.1. Business scenarios and development process

This tutorial describes an end-to-end solution for you to use big data services of Alibaba Cloud to build an online operational analytics platform based on the basic requirements for such a platform in the big data era. This solution allows you to write data to a storage system at high concurrency, process data efficiently, and analyze and display data.

Business scenarios

This tutorial uses HTTP access logs from a real website as the source data. You can analyze the access logs to obtain the following statistics:

- Page views (PVs) and unique visitors (UVs) of the website in terminal types, such as Android, iPad, iPhone, and PC.

 **Note** PV and UV are two basic metrics for measuring the traffic of a website. A PV is recorded each time a user visits a web page. If a user visits the same web page multiple times, each view is added as a PV. An UV refers to a unique visitor to a website in one day. If a user visits the website multiple times in one day, the user is counted as only one UV.

- Source regions of website traffic.

Development process

The development process in this tutorial consists of the following steps:

- Step 1: [Prepare the environment](#).
- Step 2: [Prepare data](#).
- Step 3: [Create tables](#).
- Step 4: [Design workflows](#).
- Step 5: [Configure nodes](#).
- Step 6: [Commit and test nodes](#).
- Step 7: [Display data on dashboards](#).

1.2. Prepare the environment

This topic describes how activate Tablestore, MaxCompute, DataWorks, and Quick BI before you build an online operational analytics platform.

Prerequisites

- An Alibaba Cloud account is created. If you do not have an Alibaba Cloud account, go to the [international site \(alibabacloud.com\)](https://www.alibabacloud.com) and click Free Account to create an Alibaba Cloud account.

- Your account completes real-name verification. If your account has not completed real-name verification, follow the instructions on the [Security Settings](#) page to verify your identity.

Context

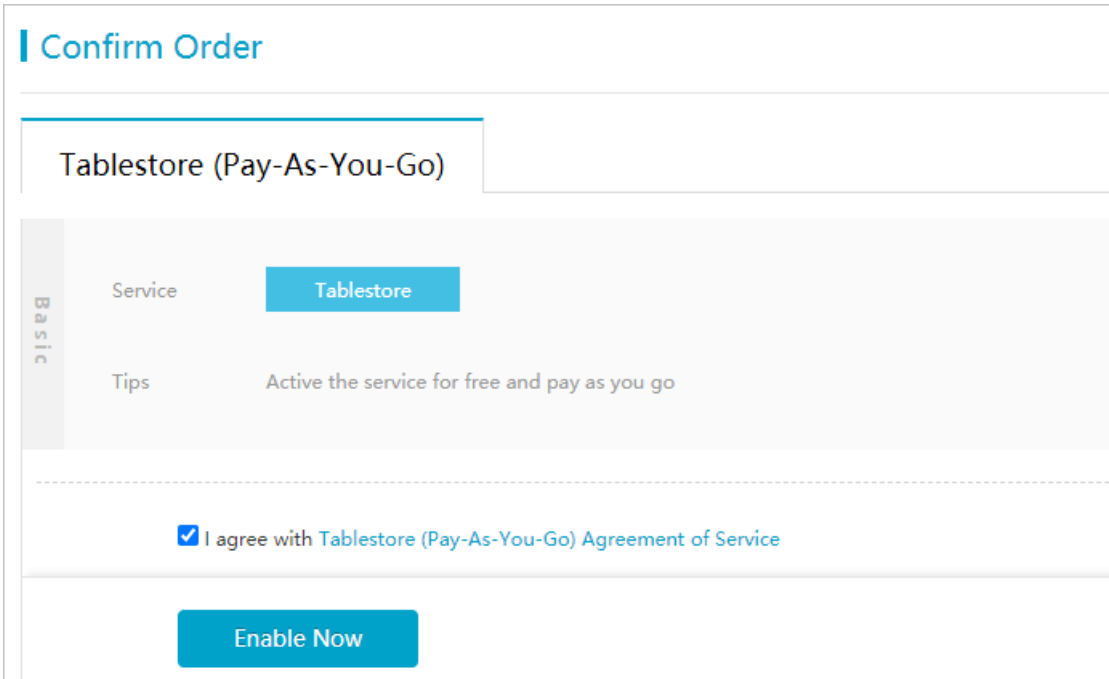
The following Alibaba Cloud services are used in this topic:

- [Tablestore](#)
- [MaxCompute](#)
- [DataWorks](#)
- [Quick BI](#)

 **Note** In this topic, Tablestore is activated in the China (Beijing) region.

Procedure

1. Activate Tablestore and create a Tablestore instance.
 - i. Go to the [Tablestore product landing page](#), and click **Get it Free** to activate Tablestore.
 - ii. On the **Confirm Order** page, read and agree to the **Tablestore (Pay-As-You-Go) Agreement of Service** by selecting the check box and click **Enable Now**.



Confirm Order

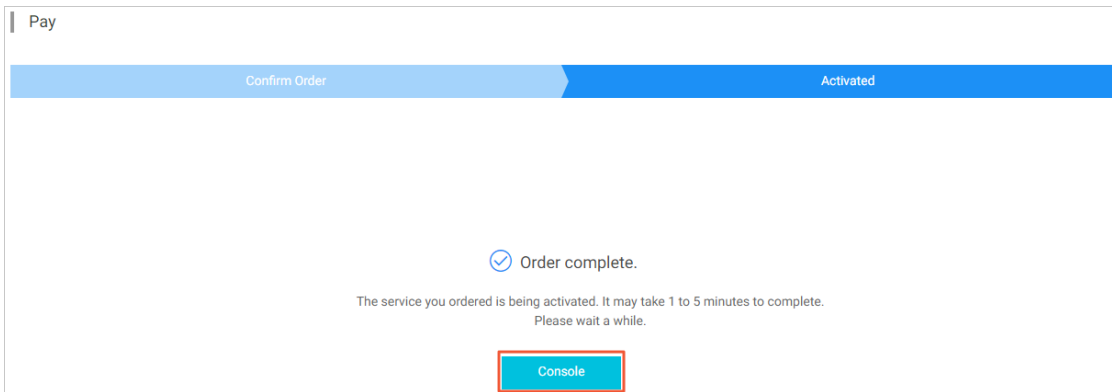
Tablestore (Pay-As-You-Go)

Basic	Service	Tablestore
	Tips	Active the service for free and pay as you go

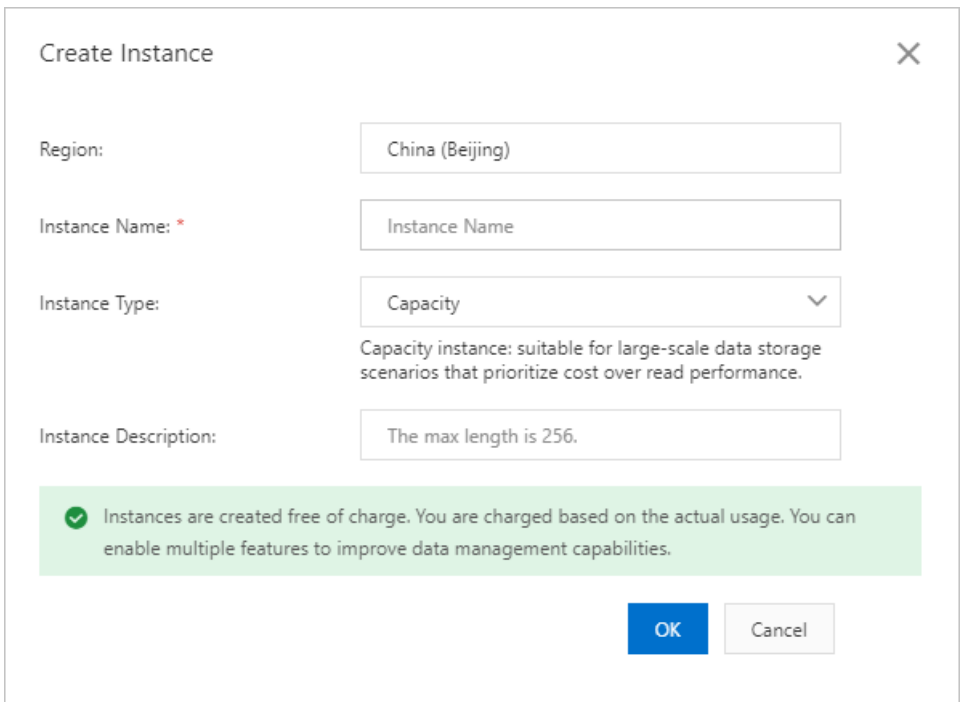
I agree with Tablestore (Pay-As-You-Go) Agreement of Service

Enable Now

iii. On the page that appears, click **Console**.



iv. On the Overview page, select **China (Beijing)** from the drop-down list at the top and click **Create Instance**. In the **Create Instance** dialog box, **Region** is automatically set to **China (Beijing)**. Enter a name in the **Instance Name** field, set **Instance Type** to **Capacity**, and then click **OK**.



Note The name of the Tablestore instance must be globally unique in each region. We recommend that you enter a name that is recognizable and complies with the naming rule. The instance name is required when you synchronize data from MaxCompute to the Tablestore instance. In this topic, the instance is named workshop-bj-001. For more information about Tablestore instances, see [Instance](#).

v. Click **All Instances** in the left-side navigation pane. On the **All Instances** page, view the created instance, which is in the **Running** status.

2. **Activate MaxCompute.**

i. Go to the [MaxCompute product landing page](#), and click **Buy Now**.

- ii. On the page that appears, set **Product Type** to **Pay-As-You-Go** and **Region** to **China (Shanghai)**. Then, click **Buy Now**.

Note The data transfer cost is reduced if you activate MaxCompute in the same region as Tablestore. Therefore, you can set **Region** to **China (Beijing)**. In this topic, the region of MaxCompute is set to **China (Shanghai)** to demonstrate the capability of using foreign tables across different regions.

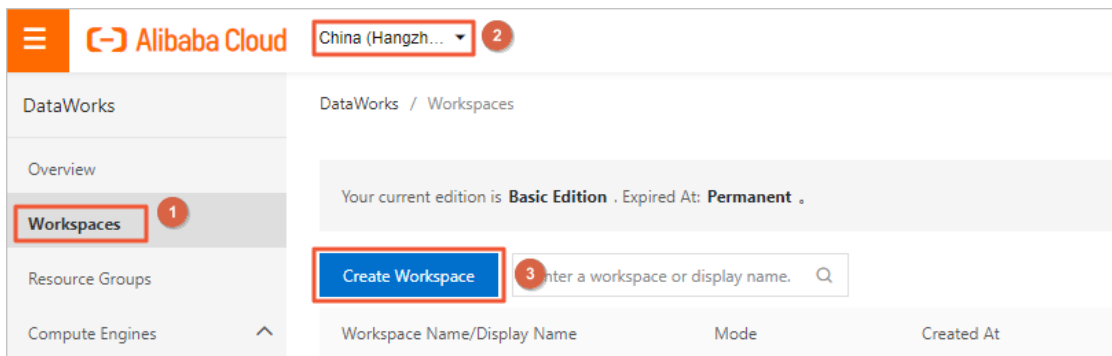
3. Activate DataWorks.

- i. Go to the **DataWorks product landing page**, and click **Buy Now**.
- ii. Set **Region** to **China (Shanghai)** and click **Buy Now**.

Note The data transfer cost can be reduced if you activate DataWorks in the same region as Tablestore. Therefore, you can set **Region** to **China (Beijing)**. In this topic, the region of DataWorks is set to **China (Shanghai)** to demonstrate the capability of accessing data across different regions.

4. Create a DataWorks workspace.

- i. Log on to the DataWorks console. Click **Workspaces** in the left-side navigation pane. On the **Workspaces** page, select **China (Hangzhou)** from the drop-down list at the top and click **Create Workspace**.



- ii. In the **Create Workspace** wizard, set parameters in the **Basic Settings** step and click **Next**.

For ease of use, the mode of the DataWorks workspace is set to **Basic Mode (Production Environment Only)** in this topic. In basic mode, a DataWorks workspace is associated with only one MaxCompute project. For more information, see [Basic mode and standard mode](#).

Create Workspace

1 Basic Settings 2 Select Engines and Services 3 Engine Details

Basic Information

* Workspace Name The specified name must start with a letter, and can only contain letters, num

Display Name The default MaxCompute project name is the same as the DataWorks workspa

* Mode ▼

Description

Advanced Settings

* Download SELECT Query Result

Note The workspace name must be globally unique. We recommend that you use a recognizable name.


- iii. In the **Select Engines and Services** step, select the required compute engines and services, and click **Next**.

In this topic, you must select **MaxCompute** as a compute engine and select **Pay-As-You-Go** as the billing method of MaxCompute.


Create Workspace


1 Basic Settings — 2 **Select Engines and Services** — 3 Engine Details

DataWorks Services


 Data Integration, Data Analytics, Operation Center, and Data Quality
Enables you to perform data synchronization and integration, orchestrate workflows, schedule and maintain recurring nodes, and check the quality of the output data.

Compute Engines

 MaxCompute Subscription [Buy Now](#) Pay-As-You-Go Developer Edition [Buy Now](#)
Allows you to develop MaxCompute SQL and MaxCompute MapReduce nodes in DataWorks.
[Go to pay](#)

 E-MapReduce
Allows you to use E-MapReduce to develop big data processing nodes in DataWorks.

Machine Learning Services

 PAI Studio Pay-As-You-Go [Buy Now](#)
Provides machine learning algorithms, deep learning frameworks, and online prediction services. To use PAI Studio, you must specify a MaxCompute computing engine.

[Next](#) [Previous](#) [Cancel](#)

- iv. In the **Engine Details** step, set parameters for the selected engines and services.

Create Workspace

✓ Basic Settings
✓ Select Engines and Services
3 Engine Details

▼ MaxCompute

* Instance display name:

* Resource Group:

* MaxCompute Data Type Edition ⓘ:

* MaxCompute Project Name ⓘ:

* Account for Accessing:

MaxCompute ⓘ

If the account used to create the MaxCompute project is a RAM sub-account, the sub-account is added to [MaxCompute Super_Administrator role](#) (Only development environment projects in standard mode).

Create Workspace
Previous
Cancel

Engine or Service	Parameter	Description
	Instance display name	The display name of the compute engine instance. The display name can be up to 27 characters in length. It must start with a letter and can contain letters, underscores (_), and digits.
	Resource Group	The quotas of computing resources and disk spaces for the compute engine instance.
	MaxCompute Data Type Edition	The data type version of the MaxCompute project.
	MaxCompute Project Name	The name of the MaxCompute project. By default, the name is the same as that of the DataWorks workspace.
MaxCompute		


Engine or Service	Parameter	Description
	Account for Accessing MaxCompute	The account for accessing the MaxCompute project. Valid values: Alibaba Cloud Account and Node Owner . If you create a DataWorks workspace in standard mode, the DataWorks workspace is associated with two MaxCompute projects, which are in the development environment and production environment, respectively. For the MaxCompute project in the development environment, the value is set to Node Owner by default. For the MaxCompute project in the production environment, we recommend that you set the value to Alibaba Cloud Account .

v. Click **Create Workspace**.

After the workspace is created, you can view information about the workspace on the **Workspaces** page.

5. Activate Quick BI.

- i. Go to the [Quick BI product landing page](#), and click **Buy Now**.
- ii. On the page that appears, set **Edition**, **Region**, **Number of Users**, and **Subscription Period** as needed. Then, click **Buy Now**.


 **Note** In the Quick BI console, you can use **Personal Workspace** or **Default Workspace**. We recommend that you use **Default Workspace**.

1.3. Prepare data

This topic describes how to use the data demo to generate data that simulates data in a real environment and can be used in subsequent data analytics.

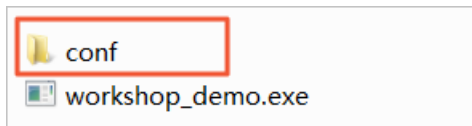
Prerequisites

- A Tablestore instance is created in the China (Beijing) region. The instance name and endpoint for connecting to the instance are obtained. You can log on to the Tablestore console and click the instance in the Instance Name column on the Overview or All Instances page to obtain the endpoint for connecting to the instance. If you connect to the instance from a region different from that of the instance, we recommend that you use the public endpoint. For more information, see [Prepare the environment](#).
- The AccessKey ID and AccessKey secret of your Alibaba Cloud account are obtained. You can log on to the Alibaba Cloud Management Console by using your Alibaba Cloud account and view the AccessKey ID and AccessKey secret on the [Security Management](#) page.

 **Note** The AccessKey ID and AccessKey secret of your Alibaba Cloud account are the credentials for accessing Alibaba Cloud APIs. Make sure that you keep your AccessKey pair safe.

Procedure

1. Download the data demo package. You can download one of the following data demo packages based on your operating system. In this topic, the Windows 7 64-bit operating system is used.
 - [Data demo package for macOS](#)
 - [Data demo package for Linux](#)
 - [Data demo package for Windows 7 64-bit](#)
2. Configure the data demo. Decompress the data demo package and edit the app.conf file in the conf directory.



The following content is in the app.conf file:

```
endpoint = "https://workshop-bj-001.cn-beijing.ots.aliyuncs.com"
instanceName = "workshop-bj-001"
accessKeyId = "LTAIF24u7g*****"
accessKeySecret = "CcwFeF3sWTPyOwsKULMw34Px*****"
usercount = "200"
daysCount = "7"
```

You must specify the following parameters:

- **endpoint:** the endpoint used to connect to the Tablestore instance. We recommend that you use the public endpoint.
 - **instanceName:** the name of the Tablestore instance.
 - **accessKeyId and accessKeySecret:** the AccessKey pair that is used to access Alibaba Cloud APIs.
3. Start the data demo to prepare test data.

- i. Start Command Prompt in Windows, go to the directory where the data demo resides, and run the following command to view the usage of commands that are related to the data demo:

```
workshop_demo.exe -h
```

The command returns the following results:

```
workshop_demo.exe -h
* prepare: Prepare test data, create data tables, and generate behavior logs of a week for users based on the user count specified in the app.conf file.
* raw ${userid} ${date} ${Top log count}: Query a specified number of logs of the specified user on the specified date.
* new/day_active/month_active/day_pv/month_pv: Query data of the following specific types in the result table. new: new users. day_active: daily active users. month_active: monthly active users. day_pv: daily page views (PVs). month_pv: monthly PVs.
```

- ii. Run the following command to prepare test data:

```
workshop_demo.exe prepare
```

The following figure shows the command output.

```
C:\Users\... \workshop_demo>workshop_demo.exe prepare
OTSObjectAlreadyExist Requested table already exists.
OTSObjectAlreadyExist Requested table already exists.
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare User data
finished one round
finished one round
finished one round
finished one round
finished one round
total insert data count is: 41757
```

In this process, the data demo automatically creates two tables in Tablestore. The following tables describe the columns in the created tables.

- o Raw log table: user_trace_log

Column	Data type	Description
md5	STRING	The first eight characters of the MD5 value of the user ID. This column is a primary key column.
uid	STRING	The user ID. This column is a primary key column.

Column	Data type	Description
ts	BIGINT	The timestamp when the user performed the operation. This column is a primary key column.
ip	STRING	The IP address of the client that sends the request.
status	BIGINT	The status code returned by the server.
bytes	BIGINT	The number of bytes returned to the client.
device	STRING	The model of the terminal used by the user.
system	STRING	The version of the operating system used by the user. The values in this column are in the format of iosxxx or androidxxx.
customize_event	STRING	The custom event, including logon, exit, purchase, registration, click, background running, user switch, and browse.
use_time	BIGINT	The use duration of the application at a time. This column is available when the custom event is exit, background running, or user switch.
customize_event_content	STRING	The content of the custom event.

o Analysis result table: analysis_result

Column	Data type	Description
metric	STRING	The metric of data. Valid values: new, day_active, month_active, day_pv, month_pv. This column is a primary key column.

Column	Data type	Description
ds	STRING	The data timestamp, in the format of yyyy-mm-dd or yyyy-mm. This column is a primary key column.
num	BIGINT	The value of the specified metric.

4. Verify data.


- Query detailed logs of a specified user.

Run the following command to query a specified number of logs of a specified user on a specified date. In the command, set the date to that when the logs are generated.

```
raw ${userid} ${date} ${Top log count}
```

In the preceding command, `${userid}` indicates the user ID, `${date}` indicates the date when the logs were generated, and `${Top log count}` indicates the number of logs to query. For example, if a table was created on June 15, 2019, you can run the `workshop_demo.exe raw 00010 "2019-06-15" 20` command to query 20 logs for the user whose ID is 00010.


```
C:\nloads\workshop_demo>workshop_demo.exe raw 00010 "2019-06-15" 20
uid      Date      bytes  customize_event
device   ip      status system
00010   2019-06-14 11:56:47 PM    759    regist
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 11:26:34 PM    252    backstage 369
iPad min2    157.249.67.241    200    ios11
00010   2019-06-14 11:21:30 PM    427    browse travel
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 11:16:03 PM    764    switch 185
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 11:06:03 PM    436    click
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 10:36:54 PM    131    click
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 10:22:26 PM    778    switch 73
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 10:06:29 PM    535    backstage 179
iPad min2    157.249.67.241    200    ios11
00010   2019-06-14 09:56:11 PM    668    click
iPad min2    157.249.67.241    200    ios11
00010   2019-06-14 09:20:45 PM    354    regist
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 09:15:37 PM    989    click
iPad min2    157.249.67.241    200    ios11
00010   2019-06-14 08:51:17 PM    460    logout 462
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 08:26:06 PM    887    comment funny
iPad min2    157.249.67.241    200    ios11
00010   2019-06-14 08:10:34 PM    278    browse finance
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 07:56:00 PM    480    click
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 07:30:11 PM    68    click
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 07:15:09 PM    398    browse news
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 07:11:21 PM    21    click
iPhone6s    222.133.108.234    200    ios10
00010   2019-06-14 06:35:07 PM    207    browse photo
iPhone7 Plus 61.103.79.217    200    ios11
00010   2019-06-14 06:24:43 PM    261    regist
iPhone7 Plus 61.103.79.217    200    ios11
```

 **Note** Tablestore is schema-free. Therefore, you do not need to pre-define attribute columns. Different events in the customize_event column have different event content. Therefore, the data demo generates both a custom event and its content in a data record.

- Query data in the analysis result table.

You can run the `workshop_demo.exe day_active` command to query the number of daily active users.

```
C:\> workshop_demo>workshop_demo.exe day_active
metric          ds              num
day_active      2019-05-19     1416104
day_active      2019-05-20     1416540
day_active      2019-05-21     1422314
day_active      2019-05-22     1422411
day_active      2019-05-23     1428480
day_active      2019-05-24     1431989
day_active      2019-05-25     1436218
day_active      2019-05-26     1437886
day_active      2019-05-27     1440633
day_active      2019-05-28     1444736
day_active      2019-05-29     1450520
day_active      2019-05-30     1451543
day_active      2019-05-31     1457510
day_active      2019-06-01     1458998
day_active      2019-06-02     1466801
day_active      2019-06-03     1468898
day_active      2019-06-04     1473173
day_active      2019-06-05     1479770
day_active      2019-06-06     1483101
day_active      2019-06-07     1484922
day_active      2019-06-08     1485347
day_active      2019-06-09     1492034
day_active      2019-06-10     1499914
day_active      2019-06-11     1495458
day_active      2019-06-12     1500697
day_active      2019-06-13     1508061
day_active      2019-06-14     1509108
day_active      2019-06-15     1510583
day_active      2019-06-16     1518355
day_active      2019-06-17     1520938
```

1.4. Build data models and analyze data

1.4.1. Create tables

Before you use MaxCompute to process data, you must create tables in MaxCompute to store both the raw data and processed data.

Prerequisites

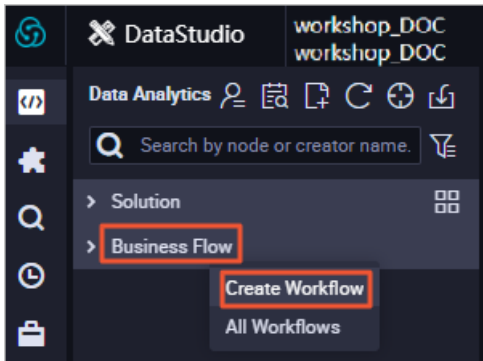
- MaxCompute is activated. A DataWorks workspace is created. A workspace in basic mode is used in this topic. For more information, see [Prepare the environment](#).
- MaxCompute is authorized to access Tablestore. If you use the same account to activate MaxCompute and Tablestore, go to the [RAM console](#) to authorize MaxCompute in one-click mode. If you use different accounts to activate MaxCompute and Tablestore, authorize MaxCompute by customizing an authorization policy. For more information, see [Access Table Store \(OTS\) data](#).

Procedure

1. Go to the DataStudio page.

- i. Log on to the DataWorks console. Click **Workspaces** in the left-side navigation pane. On the page that appears, select **China (Shanghai)** in the top navigation bar.
 - ii. Find the target workspace and click **Data Analytics** in the Actions column.
2. Create a workflow.

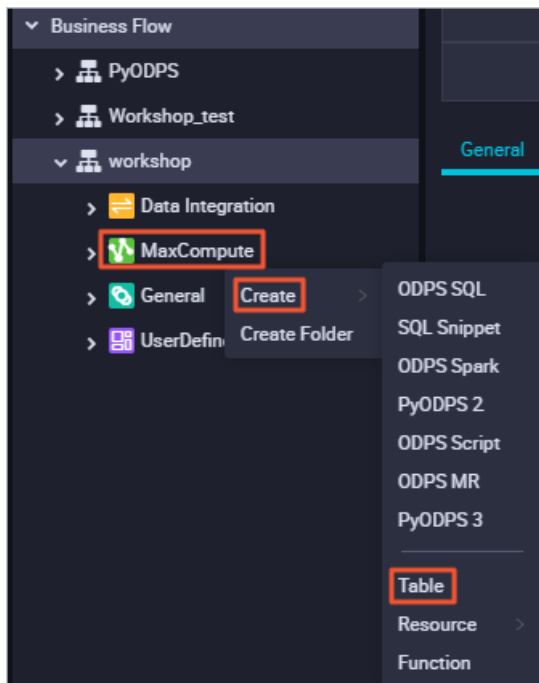
- i. On the Data Analytics tab of the DataStudio page, right-click **Business Flow** and select **Create Workflow**.



- ii. In the dialog box that appears, specify **Workflow Name** and **Description** and click **Create**. For this example, set **Workflow Name** to **Workshop**.

3. Create tables.

- i. Create an external table named `ots_user_trace_log`.
 - a. Click the **Workshop** workflow. Right-click **MaxCompute** and choose **Create > Table**. In the dialog box that appears, set **Table Name** to `ots_user_trace_log` and click **Commit**.



- b. On the tab that appears, specify **Display Name** in the **General** section and click **DDL Statement** in the upper-left corner.

The screenshot displays the configuration interface for a table named `ots_user_trace_log`. The **DDL Statement** tab is active and highlighted with a red box and a red circle containing the number 2. In the **General** section, the **Display Name** field is set to `ots_user_trace_log` and is also highlighted with a red box and a red circle containing the number 1. Other fields include **Level 1 Folder** (Select an option.), **Level 2 Folder** (Select an option.), and **Description**. The **Physical Model** section shows **Partitioning** set to **Non-Partitioned Table**, **Table Level** (Select an option.), **Table Type** set to **External Table**, **Storage Address** (tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/), and **File Format** set to **OTS**. An **Authorize** button is highlighted with a red box.

- c. In the **DDL Statement** dialog box, enter the statement that is used to create the table and click **Generate Table Schema**.

The following code describes the statement that is used to create the `ots_user_trace_log` table:

```
CREATE EXTERNAL TABLE ots_user_trace_log (  
  md5 string COMMENT 'The first eight characters in the MD5 value of the user ID',  
  uid string COMMENT 'The ID of the user',  
  ts bigint COMMENT 'The timestamp when the user performs the operation',  
  ip string COMMENT 'The IP address of the client',  
  status bigint COMMENT 'The status code returned by the server',  
  bytes bigint COMMENT 'The number of bytes returned to the client',  
  device string COMMENT 'The model of the terminal on which the client runs',  
  system string COMMENT 'The version of the operating system in which the client runs, i  
n the format of ios xxx or android xxx',  
  customize_event string COMMENT 'The custom event: logon, logoff, purchase, registrat  
ion, click, background running, user switching, browsing, or comment',  
  use_time bigint COMMENT 'The duration for which the app is used at a time. This field i  
s available when the custom event is set to logoff, background running, or user switchin  
g.',  
  customize_event_content string COMMENT 'The content of the custom event. This field  
is available when the custom event is set to browsing or comment.'  
)  
STORED BY 'com.aliyun.odps.TableStoreStorageHandler'  
WITH SERDEPROPERTIES (  
  'tablestore.columns.mapping'=':md5,:uid,:ts, ip,status,bytes,device,system,customize_  
event,use_time,customize_event_content',  
  'tablestore.table.name'='user_trace_log'  
)  
LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/';
```


- **STORED BY:** required. The value `com.aliyun.odps.TableStoreStorageHandler` specifies the storage handler that is built in MaxCompute to process the data stored in Tablestore. This clause defines the logic of interactions between MaxCompute and Tablestore.

- **SERDEPROPERITES:** required. The WITH SERDEPROPERITES clause allows you to specify options. If you specify `com.aliyun.odps.TableStoreStorageHandler` in the STORED BY clause, you must also specify the following options in the WITH SERDEPROPERITES clause:
 - **tablestore.columns.mapping:** the columns of the Tablestore table that you want to access from MaxCompute. These columns include primary key columns and property columns.


 **Note**

- To specify primary key columns, add a colon (:) to the beginning of each column name. `:md5` and `:uid` are primary key columns in this example. The remaining columns are property columns.
- If you specify column mapping by using the `tablestore.columns.mapping` option, you must specify all the primary key columns in the Tablestore table. You only need to specify the property columns that you want to access from MaxCompute. The specified property columns must exist in the Tablestore table. Otherwise, errors are returned when you query data in the external table.

- **tablestore.table.name:** the name of the Tablestore table that you want to access from MaxCompute. If you specify an invalid table name or the specified table name does not exist, an error is returned and MaxCompute does not create a Tablestore table.
- **LOCATION:** This clause specifies the endpoint of Tablestore. Specify the endpoint of your Tablestore instance in this clause. For more information, see [Prepare the environment](#).

 **Note** If you specify the public endpoint of your Tablestore instance in the LOCATION clause, such as `LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/'`, an error that indicates a network failure may be returned. In this case, specify the endpoint of the instance in a classic network, for example, `LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots-internal.aliyuncs.com/'`.

- d. Click **Commit to Production Environment**. The external table is created.

 **Note** If you create the external table in a workspace that is in standard mode, click **Commit in Development Environment** and then **Commit to Production Environment**.

- ii. Create a table named `ods_user_trace_log`. Use the same method as in the preceding example to create the table. The `ods_user_trace_log` table is a table at the operational data store (ODS) layer. The following code describes the statement that is used to create the table:

```
CREATE TABLE IF NOT EXISTS ods_user_trace_log (  
  md5 STRING COMMENT 'The first eight characters in the MD5 value of the user ID',  
  uid STRING COMMENT 'The ID of the user',  
  ts BIGINT COMMENT 'The timestamp when the user performs the operation',  
  ip STRING COMMENT 'The IP address of the client',  
  status BIGINT COMMENT 'The status code returned by the server',  
  bytes BIGINT COMMENT 'The number of bytes returned to the client',  
  device STRING COMMENT 'The model of the terminal on which the client runs',  
  system STRING COMMENT 'The version of the operating system in which the client runs, in t  
he format of ios xxx or android xxx',  
  customize_event STRING COMMENT 'The custom event: logon, logoff, purchase, registration  
, click, background running, user switching, browsing, or comment',  
  use_time BIGINT COMMENT 'The duration for which the app is used at a time. This field is av  
ailable when the custom event is set to logoff, background running, or user switching.',  
  customize_event_content STRING COMMENT 'The content of the custom event. This field is  
available when the custom event is set to browsing or comment.'  
)  
PARTITIONED BY (  
  dt STRING  
)  
);
```

- iii. Create a table named `dw_user_trace_log`. Use the same method as in the preceding example to create the table. The `dw_user_trace_log` table is a table at the data warehouse detail (DWD) layer. The following code describes the statement that is used to create the table:


```
CREATE TABLE IF NOT EXISTS dw_user_trace_log (  
  uid STRING COMMENT 'The ID of the user',  
  region STRING COMMENT 'The region in which the user resides, which is obtained based on  
the IP address',  
  device_brand string comment 'The brand of the terminal on which the client runs',  
  device STRING COMMENT 'The model of the terminal on which the client runs',  
  system_type STRING COMMENT 'The operating system in which the client runs. Valid values:  
Android, IOS, ipad, and Windows_phone.',  
  customize_event STRING COMMENT 'The custom event: logon, logoff, purchase, registration  
, click, background running, user switching, browsing, or comment',  
  use_time BIGINT COMMENT 'The duration for which the app is used at a time. This field is av  
ailable when the custom event is set to logoff, background running, or user switching.',  
  customize_event_content STRING COMMENT 'The content of the custom event. This field is  
available when the custom event is set to browsing or comment.'  
)  
PARTITIONED BY (  
  dt STRING  
)  
);
```


- iv. Create a table named `rpt_user_trace_log`. Use the same method as in the preceding example to create the table. The `rpt_user_trace_log` table is a table at the application data store (ADS) layer. The following code describes the statement that is used to create the table:

```
CREATE TABLE IF NOT EXISTS rpt_user_trace_log (  
  country STRING COMMENT 'The country in which the user resides',  
  province STRING COMMENT 'The province in which the user resides',  
  city STRING COMMENT 'The city in which the user resides',  
  device_brand string comment 'The brand of the terminal on which the client runs',  
  device STRING COMMENT 'The model of the terminal on which the client runs',  
  system_type STRING COMMENT 'The operating system in which the client runs. Valid values:  
Android, IOS, ipad, and Windows_phone.',  
  customize_event STRING COMMENT 'The custom event: logon, logoff, purchase, registration  
, click, background running, user switching, browsing, or comment',  
  use_time BIGINT COMMENT 'The duration for which the app is used at a time. This field is av  
ailable when the custom event is set to logoff, background running, or user switching.',  
  customize_event_content STRING COMMENT 'The content of the custom event. This field is  
available when the custom event is set to browsing or comment.'  
  pv bigint comment 'The number of page views (PVs)',  
  uv bigint comment 'The number of unique visitors (UVs)'  
)  
PARTITIONED BY (  
  dt STRING  
)  
);
```

4. Verify the created tables.

- i. After you create the tables, click the Workshop workflow, **MaxCompute**, and then **Table** to view the created tables.
- ii. Click the Workshop workflow and then **MaxCompute**. Right-click **Data Analytics** and choose **Create > ODPS SQL**.
- iii. In the **Create Node** dialog box, specify **Node Name** and click **Commit** to create an ODPS SQL node.

- iv. Enter the following SQL statements in the SQL editor of the ODPS SQL node and click the  icon:

```
DESCRIBE ots_user_trace_log;
DESCRIBE ods_user_trace_log;
DESCRIBE dw_user_trace_log;
DESCRIBE rpt_user_trace_log;
```

The following figure shows the schemas of the created tables.

```
+-----+
| Owner: ██████████ | Project: ██████████ |
| TableComment: |
+-----+
| CreateTime:      2020-06-16 18:56:46 |
| LastDDLTime:     2020-06-16 18:56:46 |
| LastModifiedTime: 2020-06-16 18:56:46 |
+-----+
| InternalTable: YES | Size: 0 |
+-----+
| Native Columns: |
+-----+
| Field | Type | Label | Comment |
+-----+
| country | string | | |
| province | string | | |
| city | string | | |
| device_brand | string | | |
| device | string | | |
| system_type | string | | |
| customize_event | string | | |
| use_time | bigint | | |
| customize_event_content | string | | |
| pv | bigint | | |
| uv | bigint | | |
+-----+
| Partition Columns: |
+-----+
| dt | string | |
+-----+
OK
2020-06-16 19:56:10 INFO =====
2020-06-16 19:56:10 INFO Exit code of the Shell command 0
2020-06-16 19:56:10 INFO --- Invocation of Shell command completed ---
```

1.4.2. Design workflows


You can design a workflow to arrange nodes to be run during data analytics. This topic describes how to design a workflow in which each node corresponds to a table at a data warehouse layer.

Design a workflow Zero-load node ODPS SQL node

Procedure

1. Go to the DataStudio page. On the Data Analytics tab, double-click the Workshop workflow. The canvas for editing the workflow appears on the right.
2. Drag Zero-Load Node to the canvas. In the Create Node dialog box that appears, set Node Name to start and click Commit.
3. Drag ODPS SQL to the canvas. In the Create Node dialog box that appears, set Node Name to ods_user_trace_log and click Commit. Use the same method to create another two nodes

and name them `dw_user_trace_log` and `rpt_user_trace_log`, respectively.

 **Note** The `ods_user_trace_log`, `dw_user_trace_log`, and `rpt_user_trace_log` nodes represent the operational data store (ODS), common data model (CDM), and application data store (ADS) layers of a data warehouse, respectively. For more information, see [Divide a data warehouse into layers](#).

1.4.3. Configure nodes

After you design a workflow, you need to configure each data analytics node in the workflow.

Configure nodes MaxCompute SQL Zero-load node Register a UDF


Prerequisites

A user-defined function (UDF) is registered. For more information, see [Register a UDF](#).

Register a UDF


1. Add resources.

- i. Download the [GetAddr.jar package and ip.dat file](#) for querying locations based on IP addresses. For more information about how to implement a UDF for querying locations based on IP addresses, see [Implement IP address-based location query in MaxCompute](#).
- ii. Go to the DataStudio page. On the Data Analytics tab, click the Workshop workflow and right-click **MaxCompute**. Choose **Create > Resource > File** to add a file resource, and choose **Create > Resource > JAR** to add a JAR resource.
 - When you add a file resource, upload the `ip.dat` file.
 - a. In the Create Resource dialog box, set **Resource Name**, select **Large File (over 500 KB)** and **Upload to MaxCompute**, and then click **Upload**. Select the `ip.dat` file and click **OK**.
 - b. On the tab that appears, click the **Submit** icon in the upper-left corner to commit the resource.
 - When you add a JAR resource, upload the `GetAddr.jar` package.
 - a. In the Create Resource dialog box, set **Resource Name**, select **Upload to MaxCompute**, and then click **Upload**. Select the `GetAddr.jar` package and click **OK**.
 - b. On the tab that appears, click the **Submit** icon in the upper-left corner to commit the resource.


 **Note** When you commit the resource, if a message appears and indicates inconsistent data lineage, ignore it.

2. Register a UDF.


- i. Right-click **MaxCompute** under the Workshop workflow and choose **Create > Function**. In the Create Function dialog box that appears, set **Function Name** to `getregion` and click **Commit**.

- ii. On the Register Function tab that appears, set Class Name to `odps.test.GetAddr`, Resources to `getaddr.jar,ip.dat`, and Expression Syntax to `getregion(ip string)`. Click the Save icon in the upper-left corner to save the settings and then click the  icon to commit the UDF for registration.

Configure nodes

1. Configure the zero-load node named start.
 - i. On the Data Analytics tab, double-click the Workshop workflow. In the directed acyclic graph (DAG) of the workflow, double-click the start node. The node configuration tab appears.
 - ii. Click the Properties tab. On the Properties tab that appears, click Use Root Node in the Dependencies section.
 - iii. In the Schedule section, set Rerun to Allow Regardless of Running Status.
 - iv. Click the  icon to commit the node.
2. Configure the ODPS SQL node named `ods_user_trace_log`.
 - i. In the DAG of the workflow, double-click the `ods_user_trace_log` node. On the node configuration tab that appears, enter the following SQL statements in the SQL editor:

```
insert overwrite table ods_user_trace_log partition (dt=${bdp.system.bizdate})
select
  md5,
  uid ,
  ts,
  ip,
  status,
  bytes,
  device,
  system,
  customize_event,
  use_time,
  customize_event_content
from ots_user_trace_log
where to_char(FROM_UNIXTIME(ts),'yyyymmdd')=${bdp.system.bizdate};
```

 **Note** For more information about the `${bdp.system.bizdate}` parameter, see [Scheduling parameters](#).


- ii. Click the Properties tab. On the Properties tab that appears, set Auto Parse to No in the Dependencies section.
- iii. Remove incorrect parent nodes.
- iv. Search for each correct parent node based on the workflow design and click the Add icon to add the parent node. In this example, the parent node is the start node.

- v. In the Schedule section, set Rerun to Allow Regardless of Running Status.
 - vi. Click the Submit icon.
3. Configure the ODPS SQL node named `dw_user_trace_log`. You can configure the `dw_user_trace_log` node in the same way as configuring the `ods_user_trace_log` node. Enter the following SQL statements in the SQL editor:

```
INSERT OVERWRITE TABLE dw_user_trace_log PARTITION (dt=${bdp.system.bizdate})
SELECT uid, getregion(ip) AS region
, CASE
    WHEN TOLOWER(device) RLIKE 'xiaomi' THEN 'xiaomi'
    WHEN TOLOWER(device) RLIKE 'meizu' THEN 'meizu'
    WHEN TOLOWER(device) RLIKE 'huawei' THEN 'huawei'
    WHEN TOLOWER(device) RLIKE 'iphone' THEN 'iphone'
    WHEN TOLOWER(device) RLIKE 'vivo' THEN 'vivo'
    WHEN TOLOWER(device) RLIKE 'honor' THEN 'honor'
    WHEN TOLOWER(device) RLIKE 'samsung' THEN 'samsung'
    WHEN TOLOWER(device) RLIKE 'leeco' THEN 'leeco'
    WHEN TOLOWER(device) RLIKE 'ipad' THEN 'ipad'
    ELSE 'unknown'
END AS device_brand, device
, CASE
    WHEN TOLOWER(system) RLIKE 'android' THEN 'android'
    WHEN TOLOWER(system) RLIKE 'ios' THEN 'ios'
    ELSE 'unknown'
END AS system_type, customize_event, use_time, customize_event_content
FROM ods_user_trace_log
WHERE dt = ${bdp.system.bizdate};
```

4. Configure the ODPS SQL node named `rpt_user_trace_log`. You can configure the `rpt_user_trace_log` node in the same way as configuring the `ods_user_trace_log` node. Enter the following SQL statements in the SQL editor:

```
INSERT OVERWRITE TABLE rpt_user_trace_log PARTITION (dt=${bdp.system.bizdate})
SELECT split_part(split_part(region, ',', 1), '[', 2) AS country
   , trim(split_part(region, ',', 2)) AS province
   , trim(split_part(region, ',', 3)) AS city
   , MAX(device_brand), MAX(device)
   , MAX(system_type), MAX(customize_event)
   , FLOOR(AVG(use_time / 60))
   , MAX(customize_event_content), COUNT(uid) AS pv
   , COUNT(DISTINCT uid) AS uv
FROM dw_user_trace_log
WHERE dt = ${bdp.system.bizdate}
GROUP BY uid,
   region;
```


5. Verify the configuration. On the Data Analytics tab, double-click the Workshop workflow. The workflow editing canvas appears on the right. Click the  icon in the upper-left corner. If each node is marked with a green check sign (✓), the nodes are run successfully. If a node fails to be run, right-click the node, select **View Log**, and then fix the issue based on logs.

1.4.4. Commit and test nodes

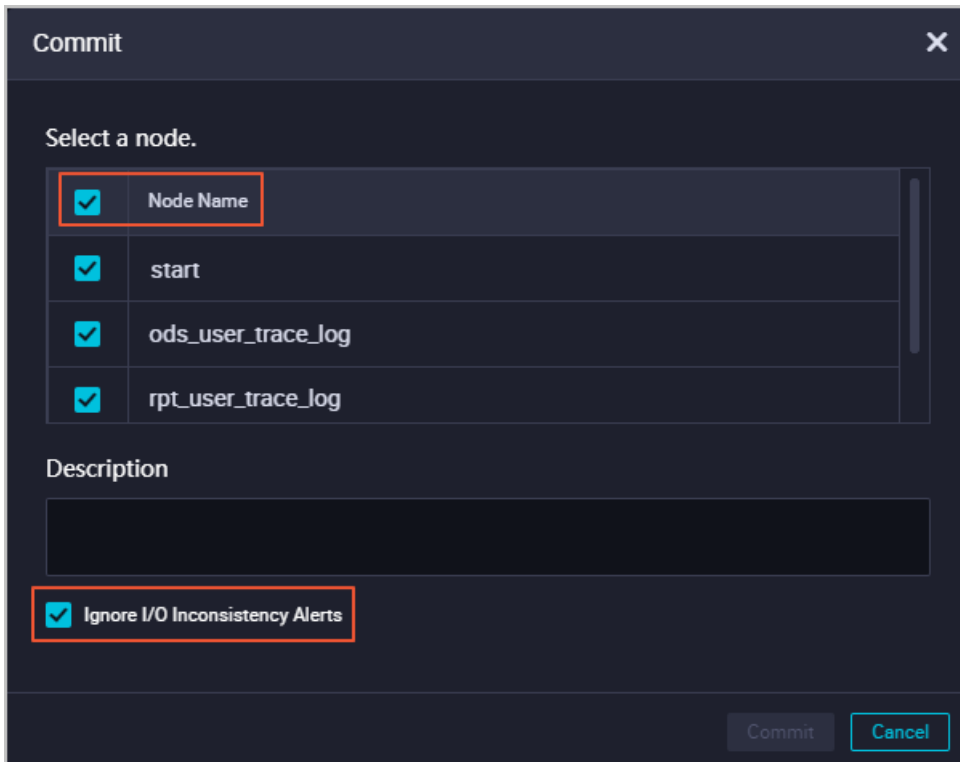
This topic describes how to commit nodes to Operation Center for testing after you configure the nodes.

commit a workflow generate retroactive data

Procedure

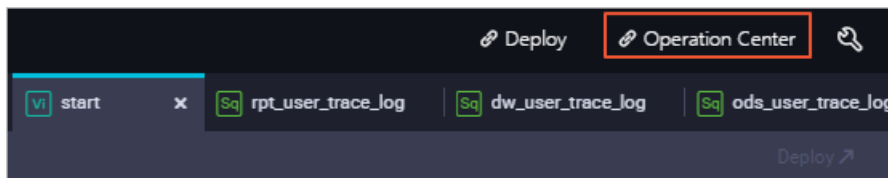
1. Optional. Commit nodes. If you have committed nodes and have not modified the nodes after node configurations, skip this step.
 - i. On the Data Analytics tab, click Business Flow and then double-click Workshop. On the Workshop tab, click the  icon.

- ii. Select all nodes that can be committed and Ignore I/O Inconsistency Alerts. Then, click Commit.

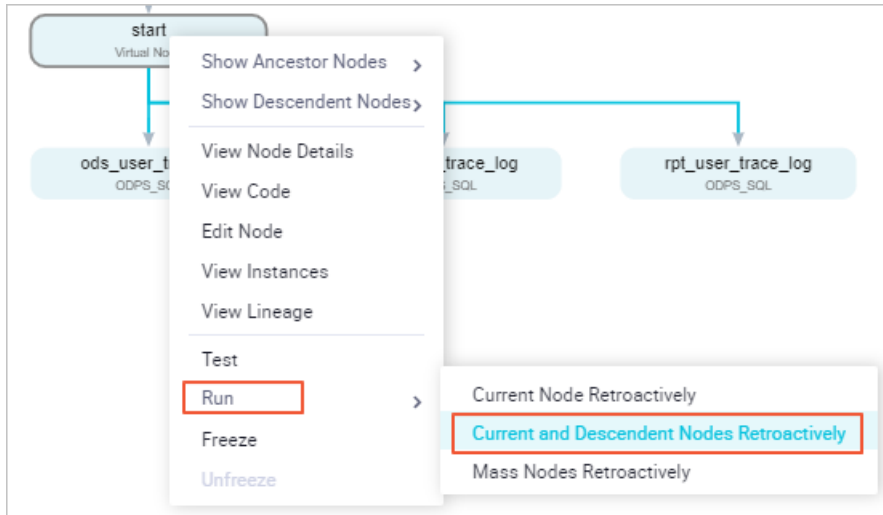


Note In a workspace in standard mode, after you commit the nodes, you must click the Deploy icon to deploy the nodes in the production environment.

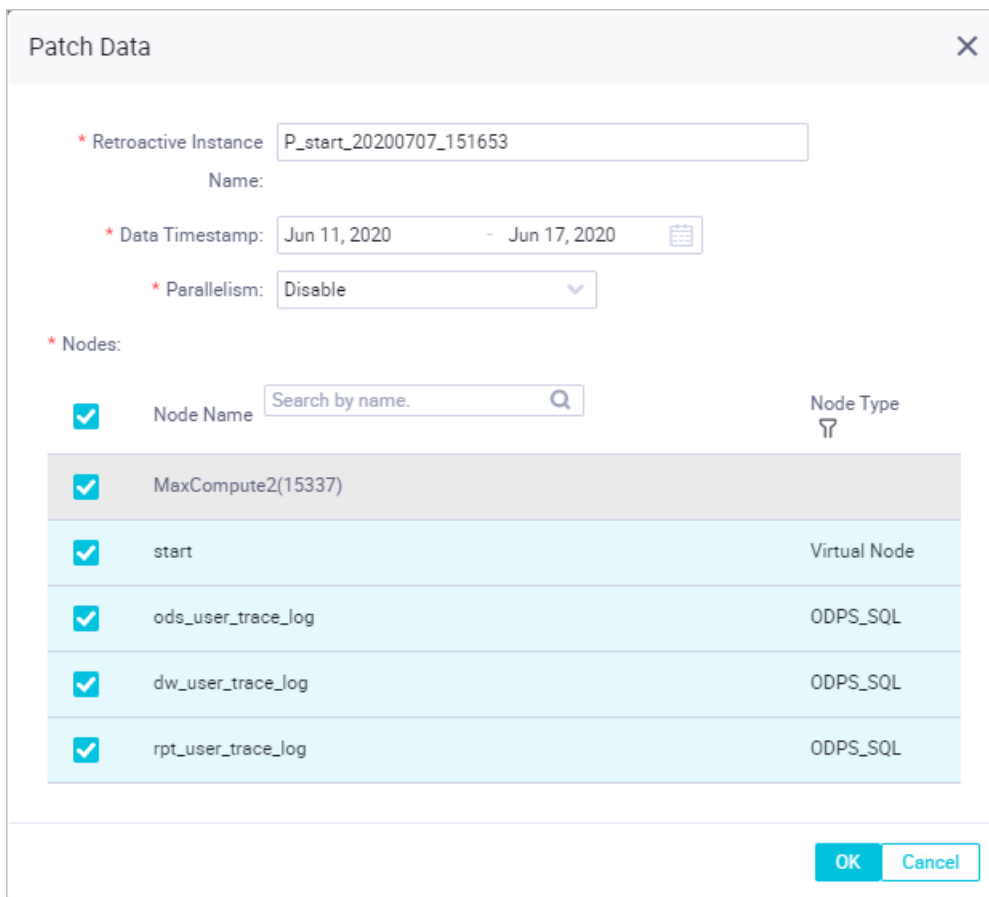
2. Click Operation Center in the upper-right corner.



3. On the page that appears, click Cycle Task Maintenance and then Cycle Task in the left-side navigation pane. On the page that appears, double-click start in the Name column.
4. In the directed acyclic graph (DAG) that appears on the right, right-click the start node and choose Run > Current and Descendant Nodes Retroactively.



5. In the **Patch Data** dialog box, select the nodes for which retroactive data needs to be generated, set the time range to the last week, and click **OK**.

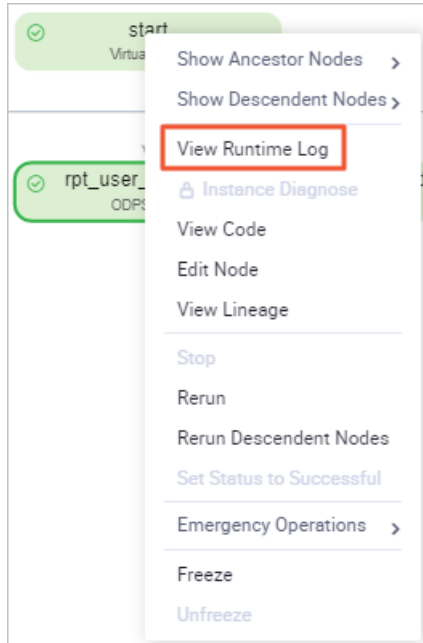


Note For more information about retroactive data generation tasks, see [Retroactive instances](#).

6. In the left-side navigation pane, click **Patch Data**. On the page that appears, view the running status of retroactive data generation tasks. You can click **Refresh** in the upper-right corner to refresh the page and view the running status of the tasks in real time.

RETROACTIVE INSTANCE NAME	STATUS	NODE TYPE	OWNER	SCHEDULE	DATA TIMESTAMP	START FROM	END AT	RUNTIME	ACTIONS
P_start_20200707_152012	Running								Stop
2020-07-06 00:00:00	Running				2020-07-06 00:00:...				
start	Successful	Virtual Node		2020-07-07 00:08:00	2020-07-06 00:00:...	2020-07-07 15:24:03	2020-07-07 15:24:03	0s	DAG Stop Rerun More

If a retroactive data generation task is abnormal, right-click the corresponding node in the DAG, select **View Runtime Log**, and then fix the issue based on the logs.

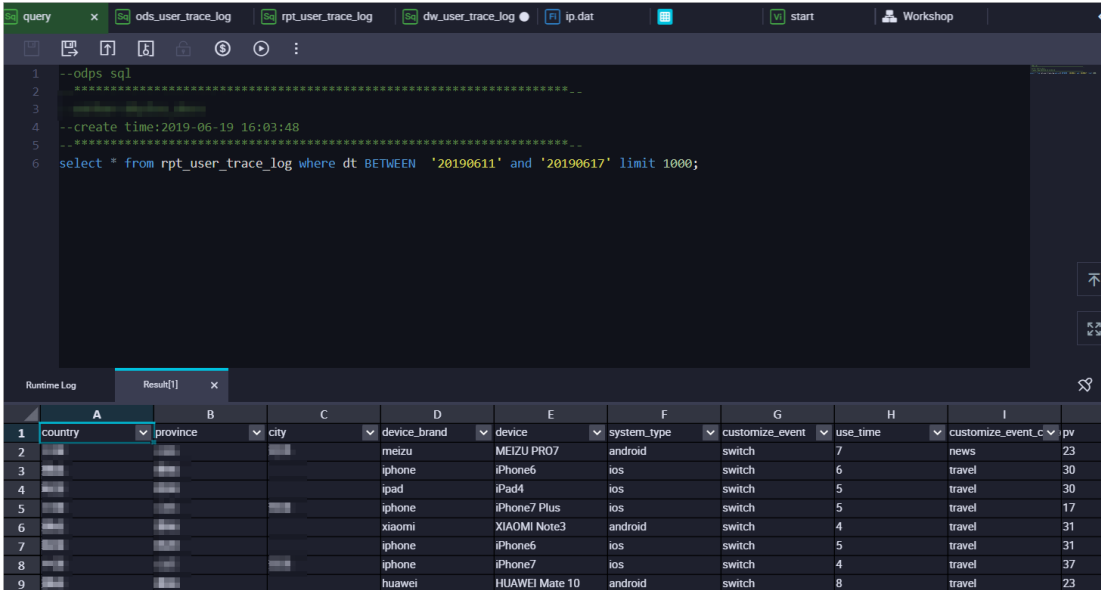


7. Verify retroactive data generation.

- i. Click Workshop and then MaxCompute. Right-click **Data Analytics** and choose **Create > ODPS SQL**. In the Create Node dialog box that appears, set Node Name to query and click Commit.
- ii. Enter the following SQL statement in the SQL editor of the node to query data from June 11, 2019 to June 17, 2019 in the rpt_user_trace_log table:

```
select * from rpt_user_trace_log where dt BETWEEN '20190611' and '20190617' limit 1000;
```

iii. Click the  icon. The following figure shows the query result.



```

1 --odps sql
2
3
4 --create time:2019-06-19 16:03:48
5
6 select * from rpt_user_trace_log where dt BETWEEN '20190611' and '20190617' limit 1000;

```

	A	B	C	D	E	F	G	H	I	
1	country	province	city	device_brand	device	system_type	customize_event	use_time	customize_event_c	pv
2				meizu	MEIZU PRO7	android	switch	7	news	23
3				iphone	iPhone6	ios	switch	6	travel	30
4				ipad	iPad4	ios	switch	5	travel	30
5				iphone	iPhone7 Plus	ios	switch	5	travel	17
6				xiaomi	XIAOMI Note3	android	switch	4	travel	31
7				iphone	iPhone6	ios	switch	5	travel	31
8				iphone	iPhone7	ios	switch	4	travel	37
9				huzwei	HUAWEI Mate 10	android	switch	8	travel	23

1.5. Display data on dashboards

After you process the `rpt_user_trace_log` table, you can create a dashboard in Quick BI to display the profiles of website users. This visualizes the data in the `rpt_user_trace_log` table.

Display data on a dashboard Quick BI

Prerequisites

An environment is prepared. Data modeling and analytics are completed. You have logged on to the [Quick BI console](#).

Context

The `rpt_user_trace_log` table contains fields such as `country`, `province`, `city`, `device_brand`, `use_time`, and `pv`. You can create a dashboard to display core metrics, periodic changes, regional distribution, and records of users.

Procedure

- In the Quick BI console, click **Default Workspace** to go to the default workspace. You can also click **Personal Workspace** to go to your personal workspace.
- Create a MaxCompute data source.
 - On the **Workspace** page, click **Data Sources** in the left-side navigation pane. The **Data Sources** page appears.
 - Click **Create Data Source** in the upper-right corner. In the dialog box that appears, click the **Cloud Data Sources** tab and click **MaxCompute**.


iii. In the **Add MaxCompute Database** dialog box that appears, set the parameters for connecting to the data source.

- **Name:** the name of the data source.
- **Database Address:** the endpoint of the data source. You can use the default endpoint.

 **Note** The endpoint of the data source varies with the region where the data source resides. For more information, see [Configure endpoints](#).

- **Project Name:** the name of the MaxCompute project.
- **AccessKey ID:** the AccessKey ID of your Alibaba Cloud account.
- **AccessKey Secret:** the AccessKey secret of your Alibaba Cloud account.

iv. Click **Test Connection** to test the connectivity to the data source.

 **Note** If the data source passes the connectivity test, a message appears and indicates that the connection is successful.

v. Click **OK**. The data source is created. After the data source is created, the **Data Sources** page appears. All tables contained in the data source are listed in the right section.

3. On the **Data Sources** page, find the `rpt_user_trace_log` table and click the **Create Dataset** icon in the **Actions** column. In the **Create Dataset** dialog box that appears, set **Name** and **Save To** and click **OK**. In this example, set **Name** to `rpt`.

4. Click **Datasets** in the left-side navigation pane. On the **Datasets** page that appears, find the created dataset and click the dataset name or the **Edit** icon in the **Actions** column. The editing page of the dataset appears.

You can perform the following operations on the dataset: change dimensions and measures, change dimension types of fields, add calculated fields, create hierarchies, change data types of fields, modify aggregate modes of measures, and create association models. For more information, see [Overview of dataset management](#).

5. Change dimension types and create hierarchies for relevant fields.

i. Change the dimension type of the `dt` field. Right-click `dt` in the left-side navigation pane and choose **Change Dimension Type > Date/Time (Source Format) > yyyyMMdd**.

ii. Change the dimension types of fields that specify geographic information.

a. Right-click `province` in the left-side navigation pane and choose **Change Dimension Type > Geo > State/Province/Municipality**.

b. Right-click `city` in the left-side navigation pane and choose **Change Dimension Type > Geo > City**. After you change the dimension types of the province and city fields, a location icon appears before the fields in the left-side navigation pane.

iii. Create a hierarchy.

a. Right-click `province` in the left-side navigation pane and select **Create Hierarchy**. In the **Create Hierarchy** dialog box that appears, click **OK**.

b. Move the `city` field and place it under the `province` field in the created hierarchy.

c. Click **Save** and return to the **Datasets** page.

6. Create a dashboard. You can create a dashboard to display the latest data as the data

changes.

- i. On the **Datasets** page, find the **rpt** dataset, and click the **Create Dashboard** icon in the **Actions** column. On the page that appears, select **Standard**. The dashboard editing page appears.
- ii. Create two kanbans. Drag the **Kanban** icon at the top to the canvas twice.
 - **Kanban 1:** On the **Data** tab, select the **rpt** dataset from the drop-down list at the top and drag **pv** to the **Indicators / Measures** section. The **rpt_user_trace_log** table is a partitioned table. You must select a dimension under **dt**, such as **dt(year)**, drag it to the **Filters** section, click the **Filter** icon next to the dimension, and then specify a time period in the **Set Filter** dialog box. In this example, the specified time period is 2019 to 2019. Then, click **Update** at the bottom of the **Data** tab.
 - **Kanban 2:** On the **Data** tab, select the **rpt** dataset from the drop-down list at the top and drag **uv** to the **Indicators / Measures** section. Complete other operations in the same way as configuring Kanban 1. Click **Update** at the bottom of the **Data** tab. You can set the display name of the kanbans on the **Style** tab.
- iii. Create a trend chart. Drag the **Line Chart** icon at the top to the canvas. Set parameters on the **Data** tab and click **Update**.
 - **Value Axis (Mea.):** Drag **pv** and **uv** to this section.
 - **Category Axis (Dim.):** Drag **dt(day)** to this section.
 - **Filters:** Drag **dt(year)** to this section.
- iv. Create a geo map. Click the **Color map** icon at the top. On the **Data** tab, select the **rpt** dataset from the drop-down list at the top. Drag **province** to the **Geo Location (Dim.)** section and **pv** to the **Colorscale (Mea.)** section. Click **Update** at the bottom of the **Data** tab.
- v. Click **Save** and **Preview** in the upper-right corner in sequence to view the created dashboard.