Batch Compute

Best Practices

Product Documentation

Contents

Best Practices
Example: 3ds Max 2018 Rendering

Last updated : 2020-07-14 12:29:09

Quick Start

This document describes how to submit a job in the BatchCompute Console to render and export a 3ds Max 2018 image.

Step 1. Make a custom image

1. Create a custom image.
2. Install 3ds Max 2018 as instructed by the official website.

> - Please temporarily disable the Windows Firewall to avoid blocking software downloads. For more information on how to select an appropriate graphics card type to avoid graphics card initialization failure, please see Display Driver Selection Dialog. In normal circumstances, you are recommended to select "Nitrous Software".

Step 2. Prepare the rendering files

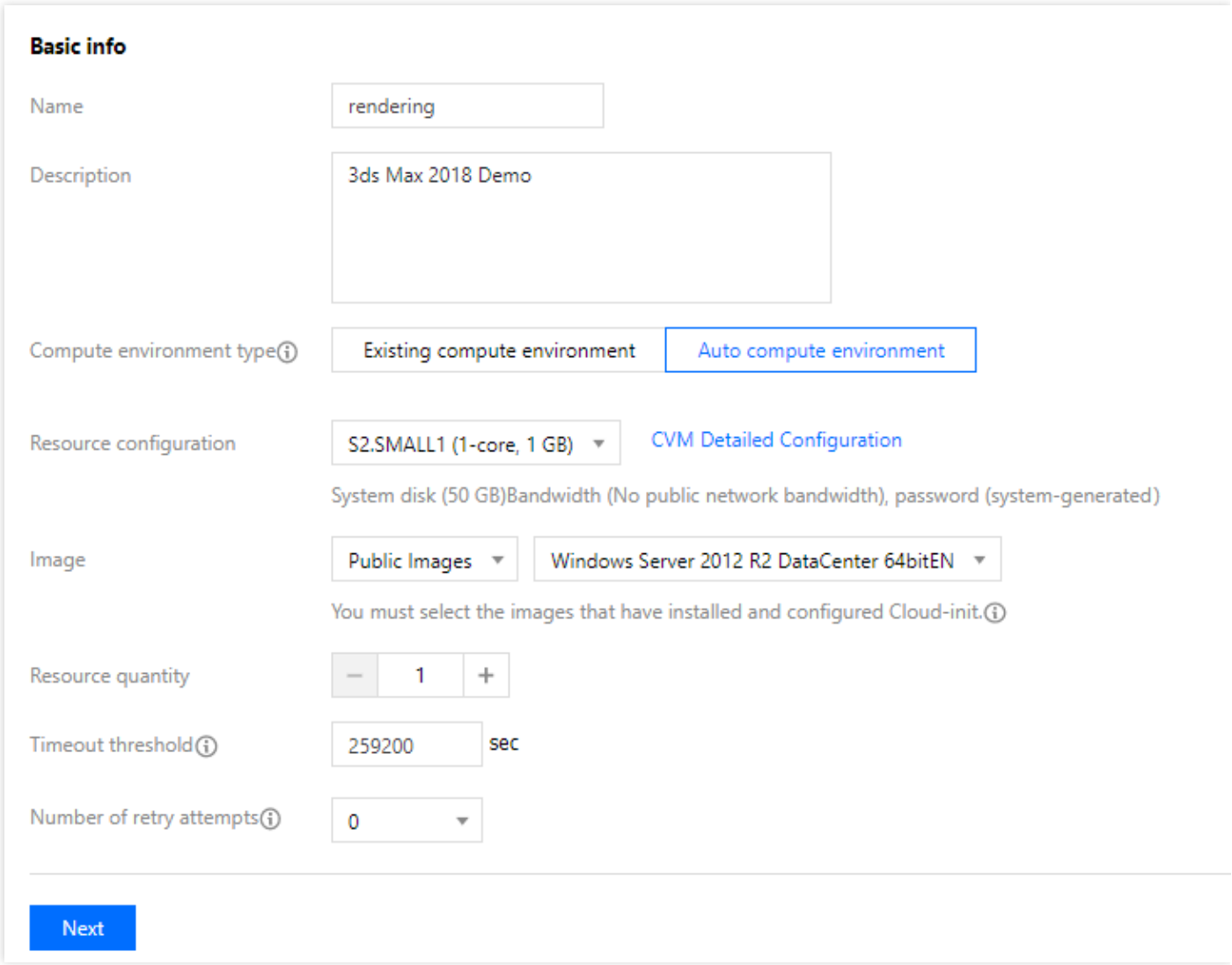
There are two mainstream ways to store the rendering materials: Cloud Object Storage (COS) and Cloud File Storage (CFS). By configuring the mount parameters, BatchCompute will mount COS or CFS to the local storage before the rendering job runs, so the renderer can access COS or CFS as if it were a local file.

- If the rendering materials are small, you are recommended to compress them into a gzip package and upload the package to COS. For more information, please see Uploading Object.
- If the rendering materials are large, you are recommended to store them in CFS.

Step 3. Create a task template

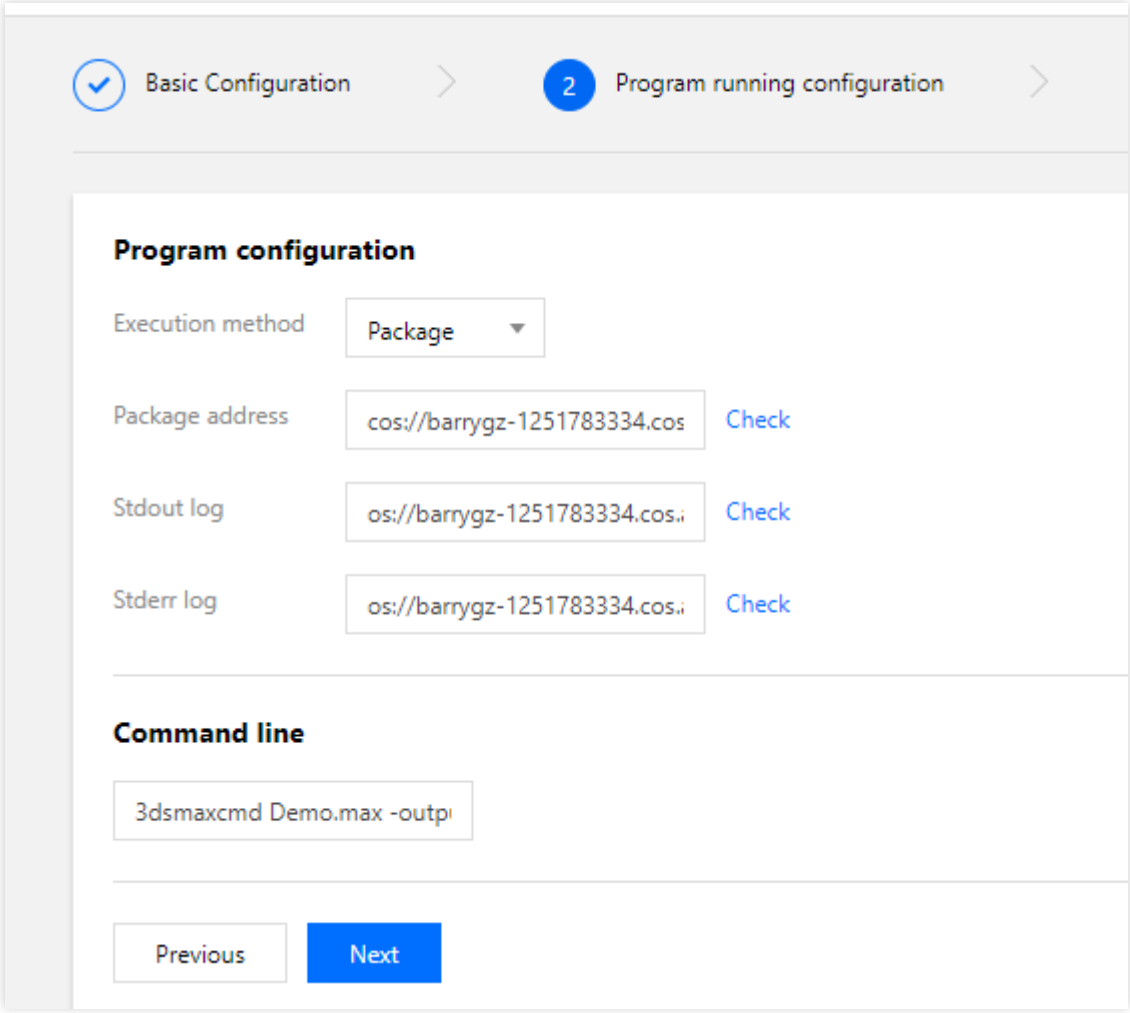
1. Log in to the BatchCompute Console and select **Task Template** on the left sidebar.
2. Select the target region at the top on the "Task Template" page and click **Create**.

3. Click **Create** to enter the "Create Task Template" page and create a template as shown below:



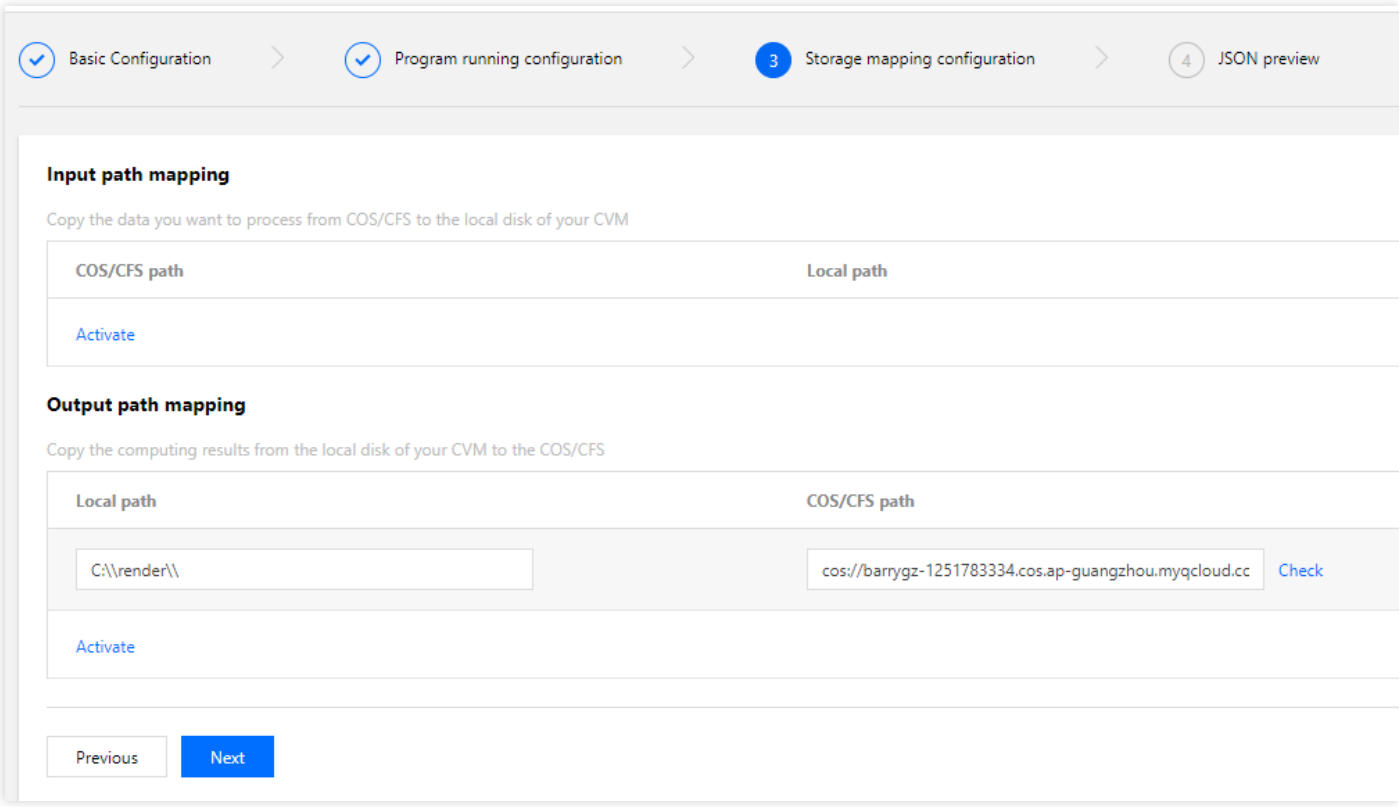
* **Name**: custom name, such as `rendering`.
* **Description**: custom description, such as `3ds Max 2018 Demo`.
* **Compute Environment Type**: select as needed. **Automatic Compute Environment** is selected in this example.
* **Resource Configuration**: S1.LARGE8 (4-core 8 GB).
* **Image**: custom image ID. Please select one as needed.
* **Number of Resources**: number of concurrent renderings, such as 1
* **Timeout Period and Number of Retries**: keep the default values.

4. Click **Next** to configure the program information as shown below:



- **Execution Method**: PACKAGE.
  * **Package Address**: COS as an example, `cos://barrygz-1251783334.cos.ap-guangzhou.myqcloud.com/render/max.tar.gz` .
  * **Stdout Log**: for more information on the format, please see How to Enter COS and CFS Paths.
- **Stderr Log**: same as Stdout log.
- **Command Line**: `3dsmaxcmd Demo.max -outputName:c:¥¥render¥¥image.jpg` .

5. Click **Next** to set the storage mapping as shown below:

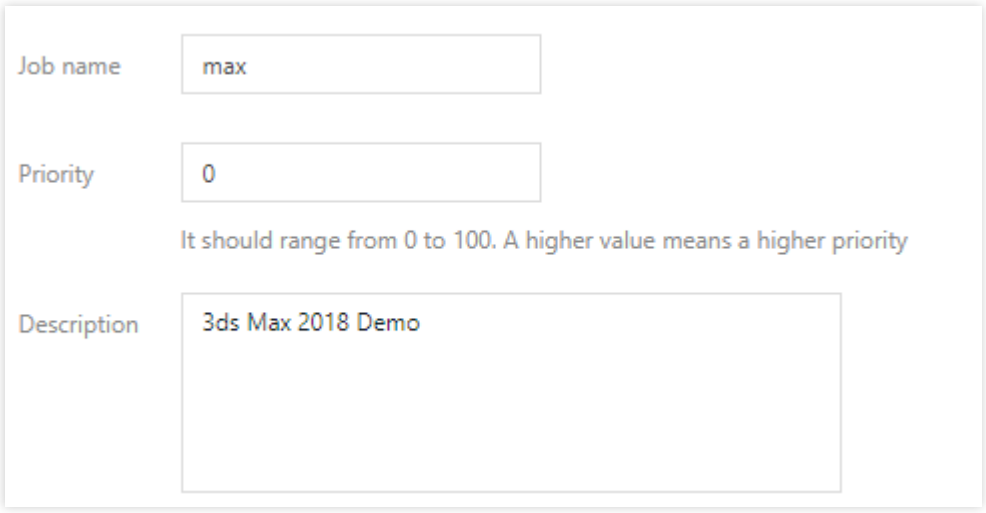


   * **Output path mapping - local path**: `C:¥¥render¥¥` .
   * **Output path mapping - COS or CFS path**: for formats, please see Entering COS or CFS Path.
6. Click **Next** to preview the task's JSON file.
7. After confirming that everything is correct, click **Save**.

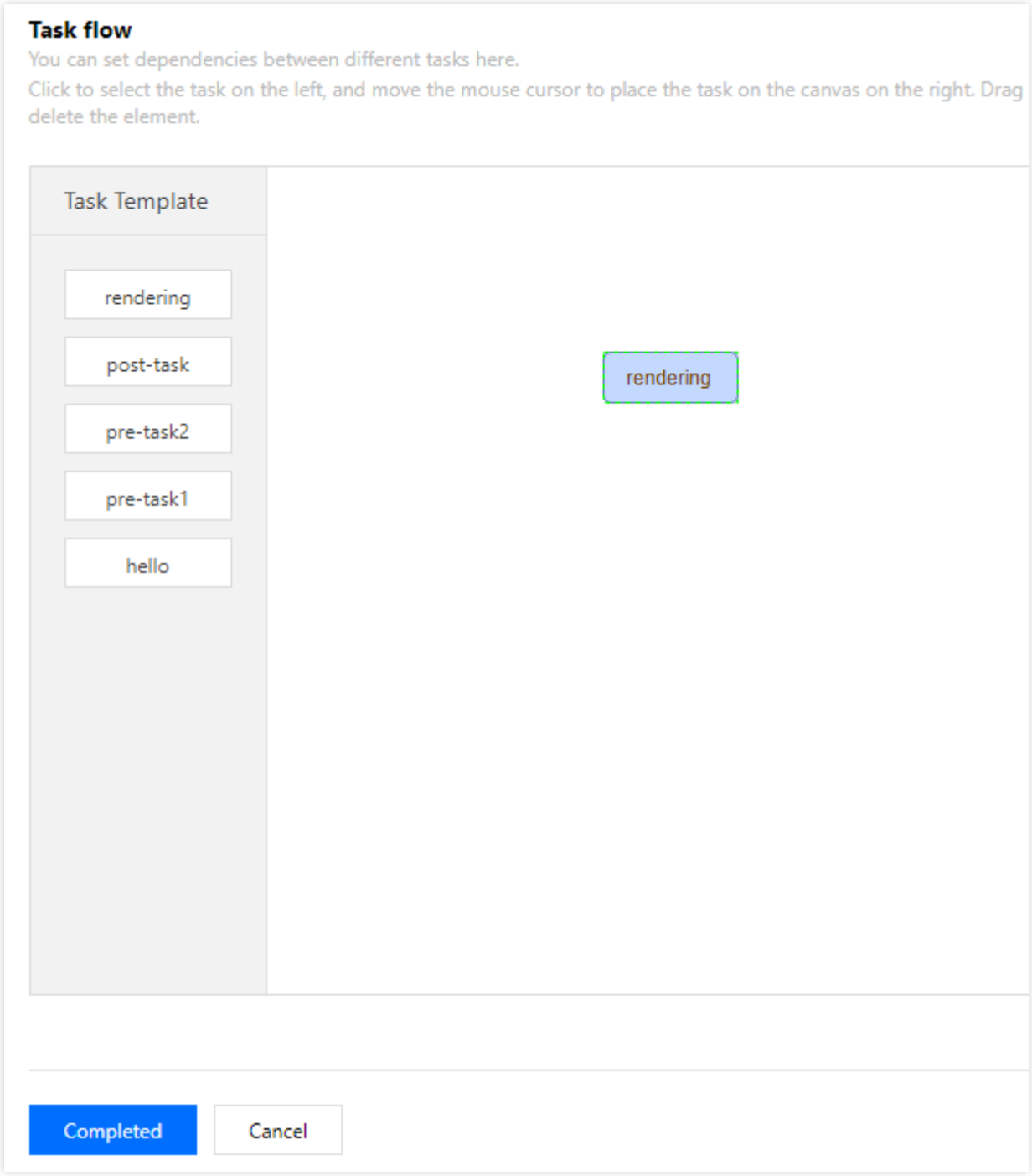
Step 4. Submit a job

1. Click **Job** on the left sidebar to enter the "Job" list page.
2. Select the target region at the top of the page and click **Create**.

3. In the "Create Job" window, configure basic information of the job as shown below:

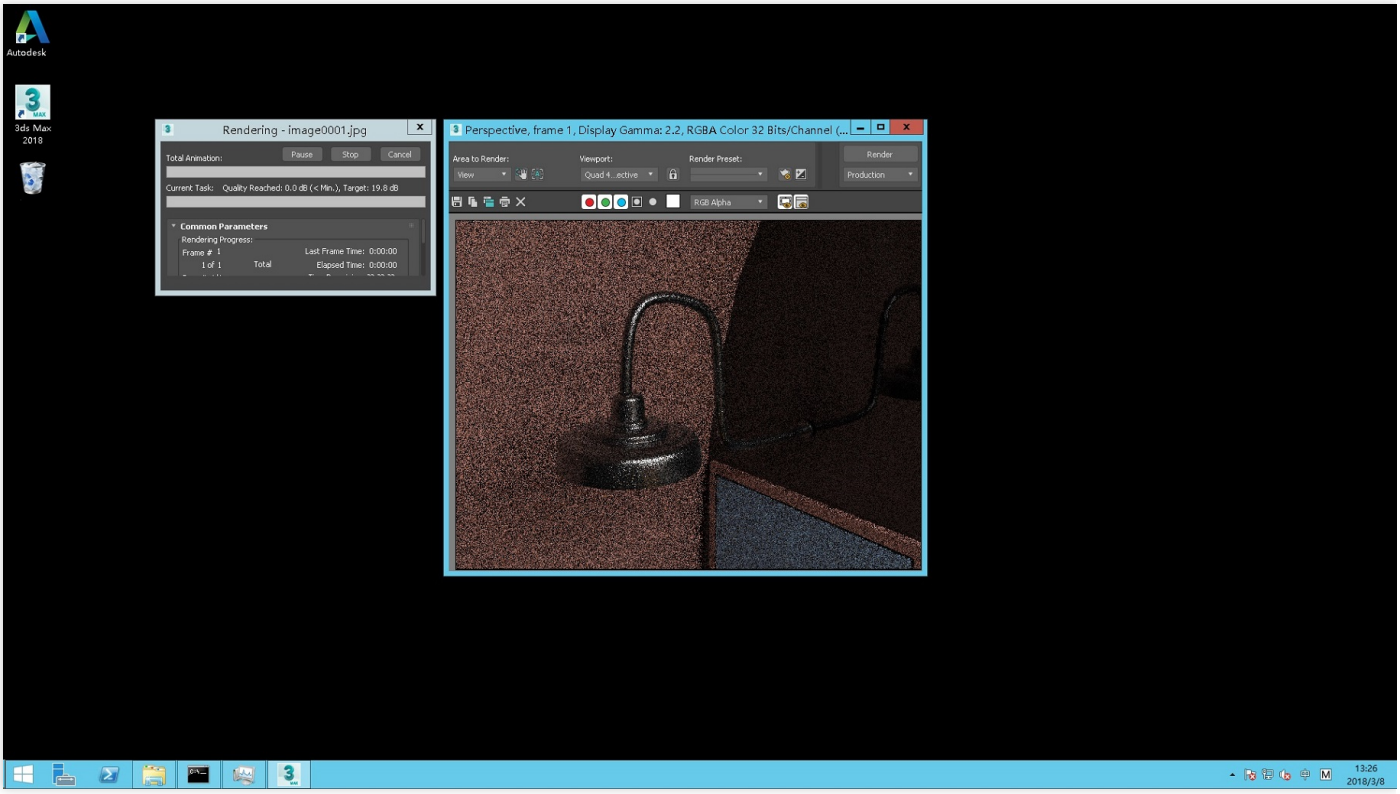


   * **Job Name**: `max` .
   * **Priority**: default value.
   - **Description**: `3ds Max 2018 Demo` .
4. Select the **rendering** task on the left on the task flow page and drag the task to the canvas on the right.



5. Enable **Task Details** on the right on the task flow page, confirm that the configuration is correct, and click **Complete**.
6. Query the job running information. For more information, please see Querying Information.

7. Below is the rendering process demonstration.



8. Query the rendering result. For more information, please see Viewing Object Information.

Subsequent Operations

This document illustrates a simple single-instance rendering job to show the most basic capabilities of BatchCompute. You can continue to test the advanced capabilities of BatchCompute as instructed in the Console User Guide.

- **Various CVM configurations**: BatchCompute provides a variety of CVM configuration options. You can customize your own CVM configuration based on your business scenario.
- **Remote storage mapping**: BatchCompute optimizes storage access and simplifies access to remote storage services into operations in the local file system.
- **Concurrent multi-image rendering**: BatchCompute supports specifying the number of concurrent rendering instances which are distinguished between through environment variables with each instance reading a different material for concurrent rendering.

Example: Deep Learning

Last updated : 2020-12-09 09:13:11

Quick Start

This document describes how to write a multilayer perceptron (MLP) BP algorithm based on a Scikit-learn machine learning library to predict the probability of winning and losing between two football teams by modeling historical international football matches, team rankings, physical and skill metrics of players, and the FIFA 2018 group match results. Below are the detailed directions.

Step 1. Make a custom image

1. For creation steps, please see Creating Custom Images.
2. Install the dependency package. Take CentOS 7.2 64-bit as an example:

```
yum -y install gcc
yum -y install python-devel
yum -y install tkinter
yum -y install python-pip
pip install --upgrade pip
pip install pandas
pip install numpy
pip install matplotlib
pip install seaborn
pip install sklearn
pip install --upgrade python-dateutil
```

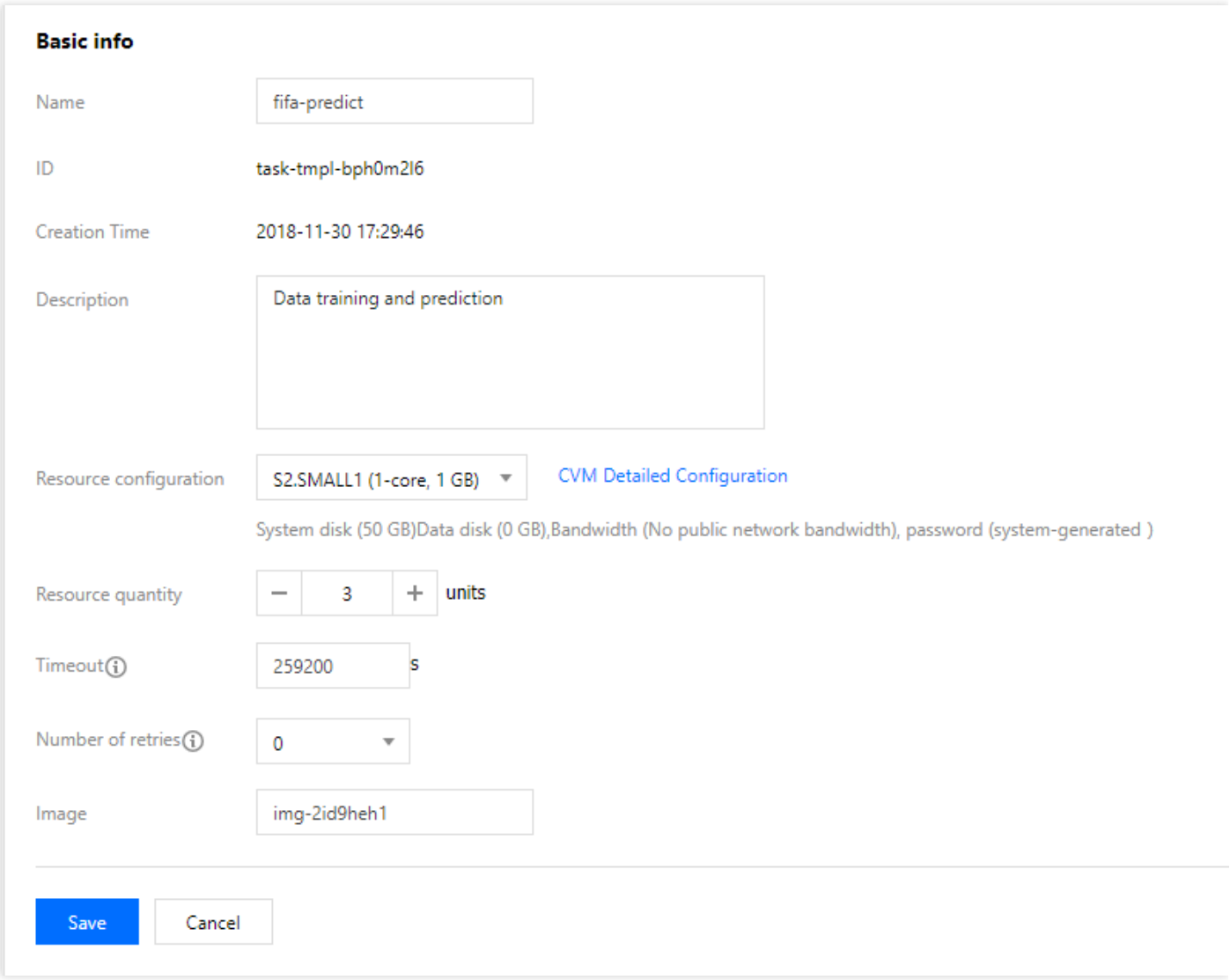
Step 2. Download the application package

Click here to download the application package and upload it to COS. Specify the COS endpoint of the package, BatchCompute will download the package to the CVM instance before the job starts and automatically decompress and execute it.

Step 3. Create a "fifa-predict" task template

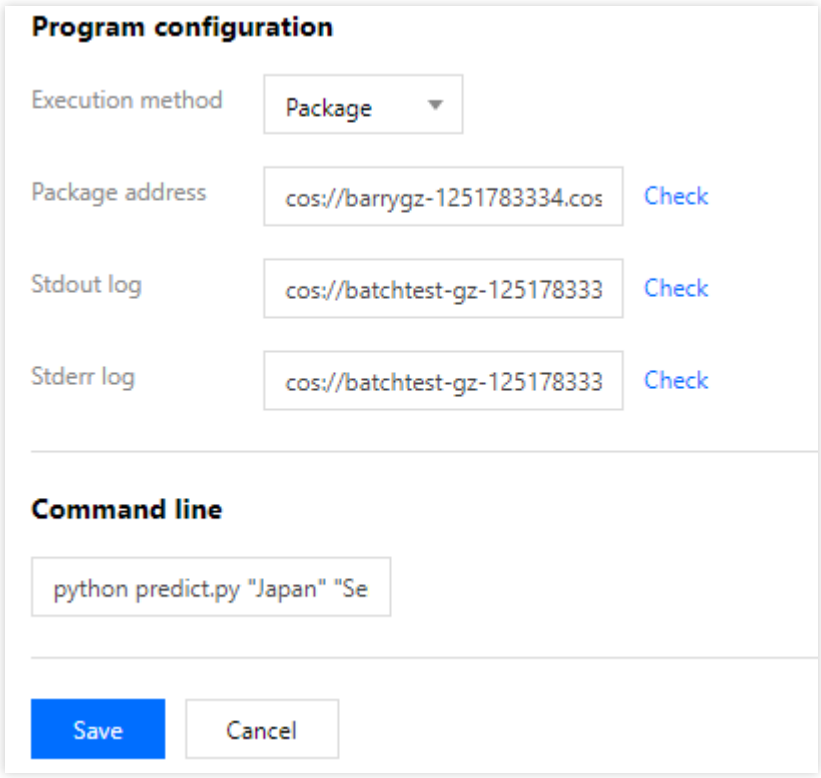
1. Log in to the BatchCompute Console and select **Task Template** on the left sidebar.
2. Select the target region at the top on the "Task Template" page and click **Create**.

3. Click **Create** to enter the "New task template" page and create a template as shown below:



- **Name**: fifa-predict.
- **Description**: data training and prediction.
- **Compute environment type**: select as needed. **Auto compute environment** is selected in this example.
- **Resource configuration**: S2.SMALL1 (1 core 1 GB memory). Public network bandwidth is pay-as-you-go.
- **Image**: custom image ID. Please select the one created in step 1.
- **Resource quantity**: number of concurrent renderings, such as 3, which means to train 3 neural network models concurrently.
- **Timeout threshold and number of retry attempts**: keep the default values.

4. Click **Next** to configure the program information as shown below:



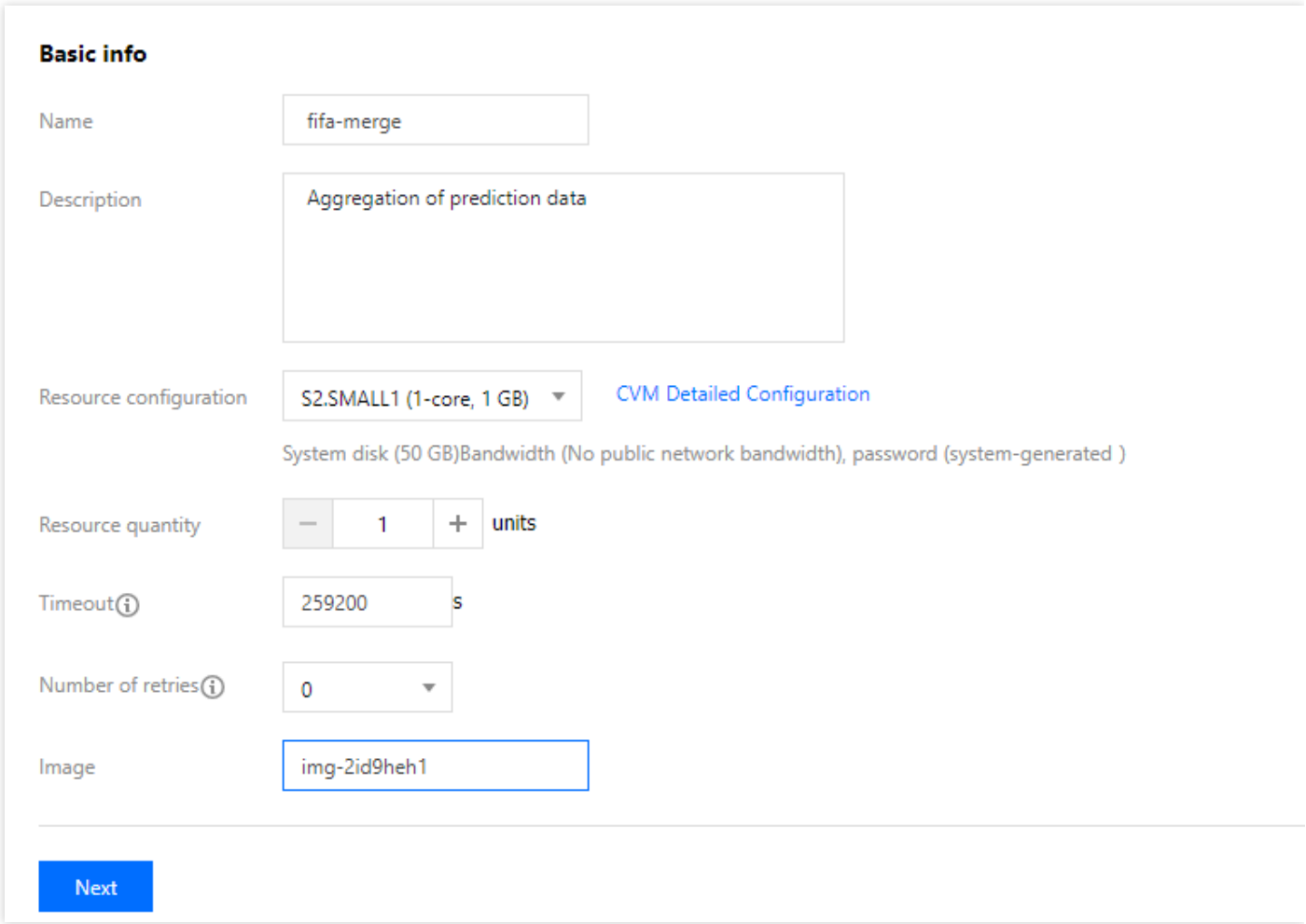
- **Execution method**: PACKAGE.
- **Package address**: using COS as an example: `cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz` .
- **Stdout log**: for more information on the format, please see Entering COS & CFS Paths.
- **Stderr log**: same as Stdout log.
- **Command line**: `python predict.py ”Japan” ”Senegal”` .
  Team list: 'Russia', 'Saudi Arabia', 'Egypt', 'Uruguay', 'Portugal', 'Spain', 'Morocco', 'Iran', 'France', 'Australia', 'Peru', 'Denmark', 'Argentina', 'Iceland', 'Croatia', 'Nigeria', 'Brazil', 'Switzerland', 'Costa Rica', 'Serbia', 'Germany', 'Mexico', 'Sweden', 'Korea Republic', 'Belgium', 'Panama', 'Tunisia', 'England', 'Poland', 'Senegal', 'Colombia', 'Japan'.

5. Skip the storage mapping configuration step and click **Next**.
6. Preview the task's JSON file and click **Save** after confirming that everything is correct.

Step 4. Create a "fifa-merge" task template

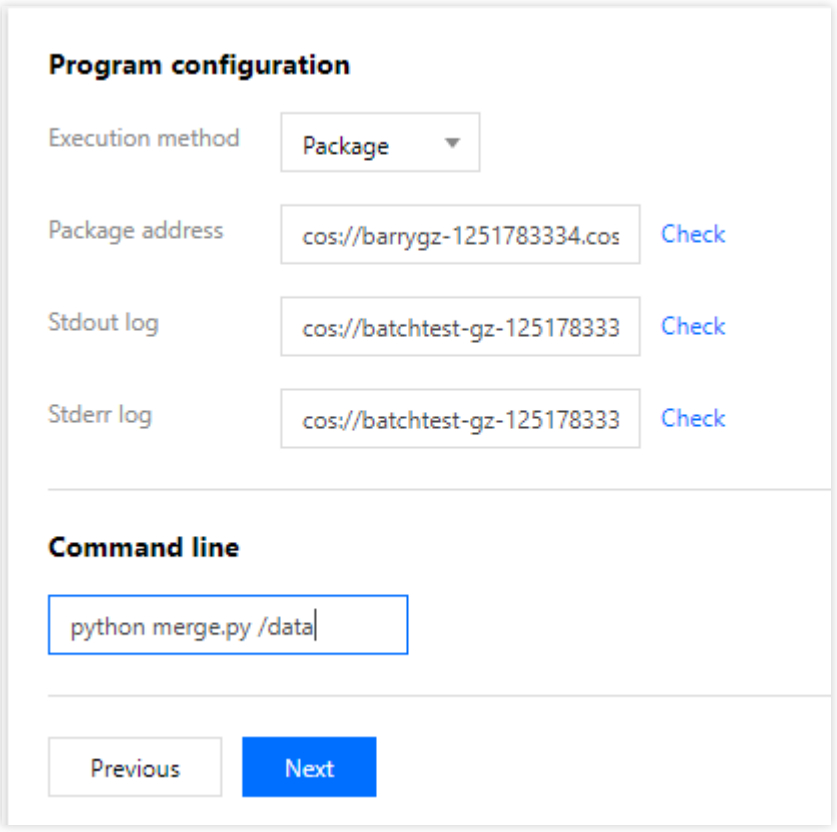
1. Log in to the BatchCompute Console and select **Task Template** on the left sidebar.
2. Select the target region at the top on the "Task Template" page and click **Create**.

3. Click **Create** to enter the "New task template" page and create a template as shown below:



- **Name**: fifa-merge.
- **Description**: aggregation of prediction data.
- **Compute environment type**: select as needed. **Auto compute environment** is selected in this example.
- **Resource configuration**: S2.SMALL1 (1 core 1 GB memory). Public network bandwidth is pay-as-you-go.
- **Image**: custom image ID. Please select the one created in step 1.
- **Resource quantity**: 1.
- **Timeout threshold and number of retry attempts**: keep the default values.

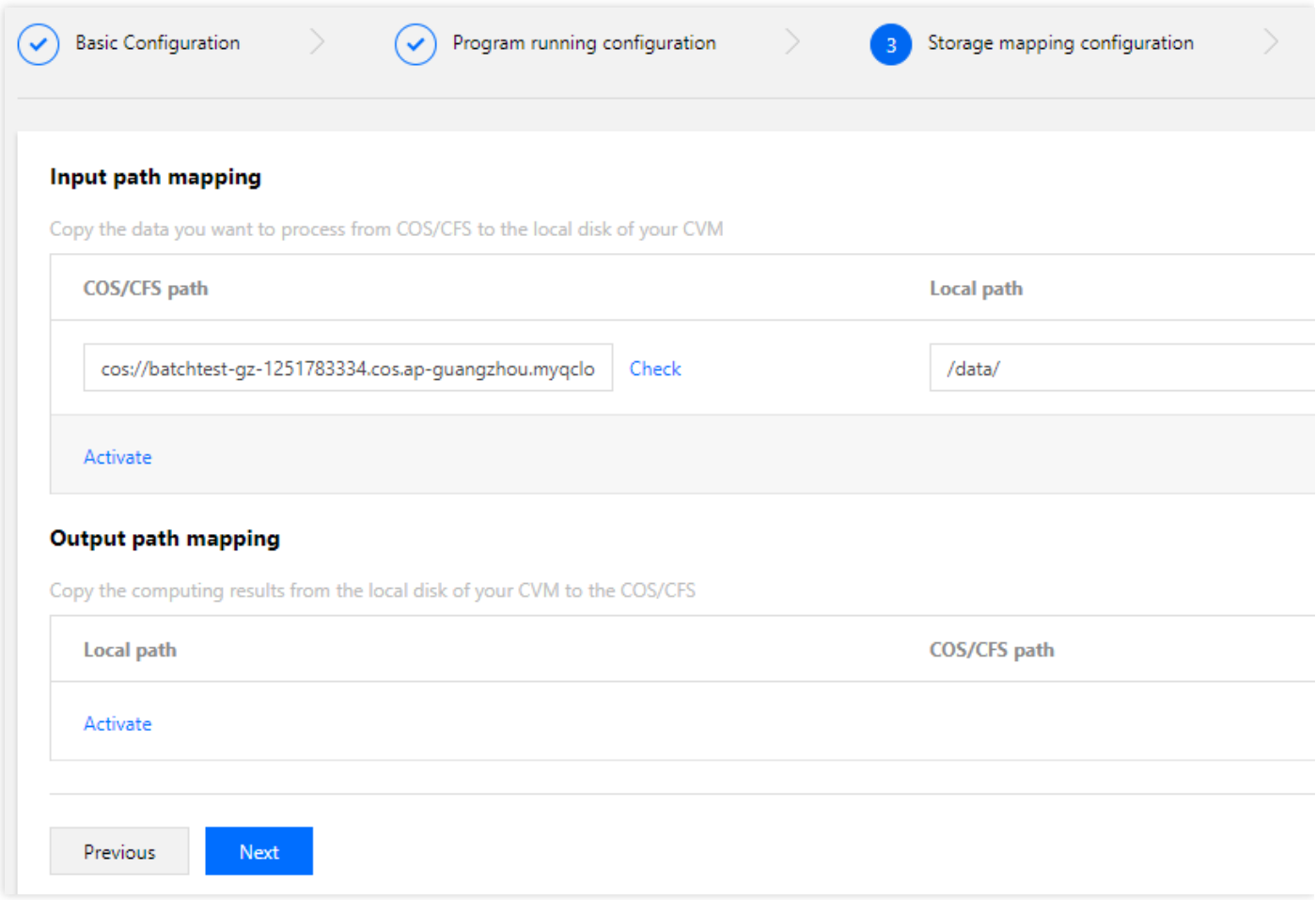
4. Click **Next** to configure the program information as shown below:



)

- **Execution method**: PACKAGE.
- **Package address**: using COS as an example: `cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz` .
- **Stdout log**: for more information on the format, please see Entering COS & CFS Paths.
- **Stderr log**: same as Stdout log.
- **Command line**: `python merge.py /data` .

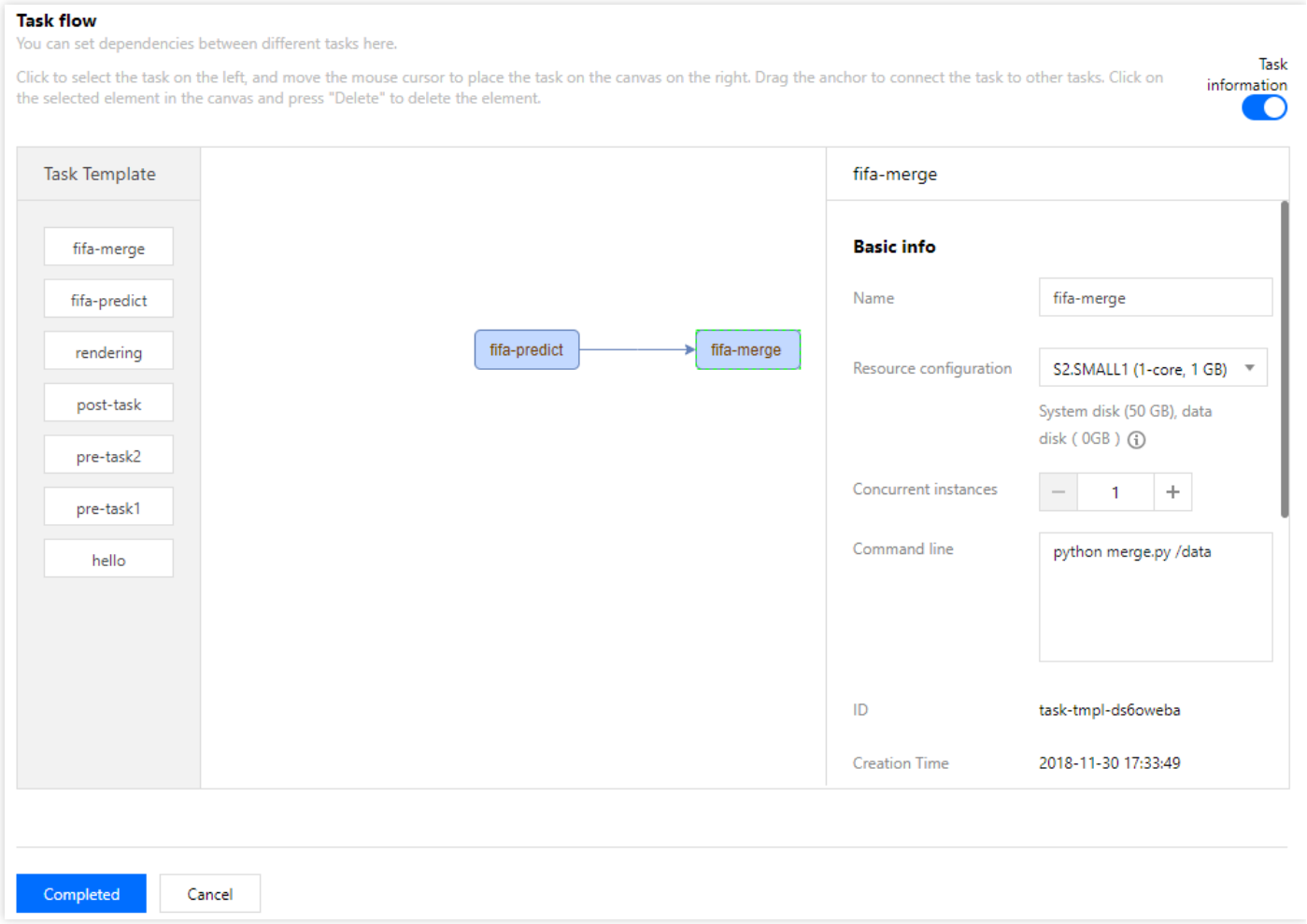
5. Click **Next** to configure the storage mapping as shown below:



   - **Input path mapping > COS/CFS path**: enter the Stdout log path of the "fifa-predict" template.
   - **Input path mapping > Local path**: `/data` .
6. Preview the task's JSON file and click **Save** after confirming that everything is correct.

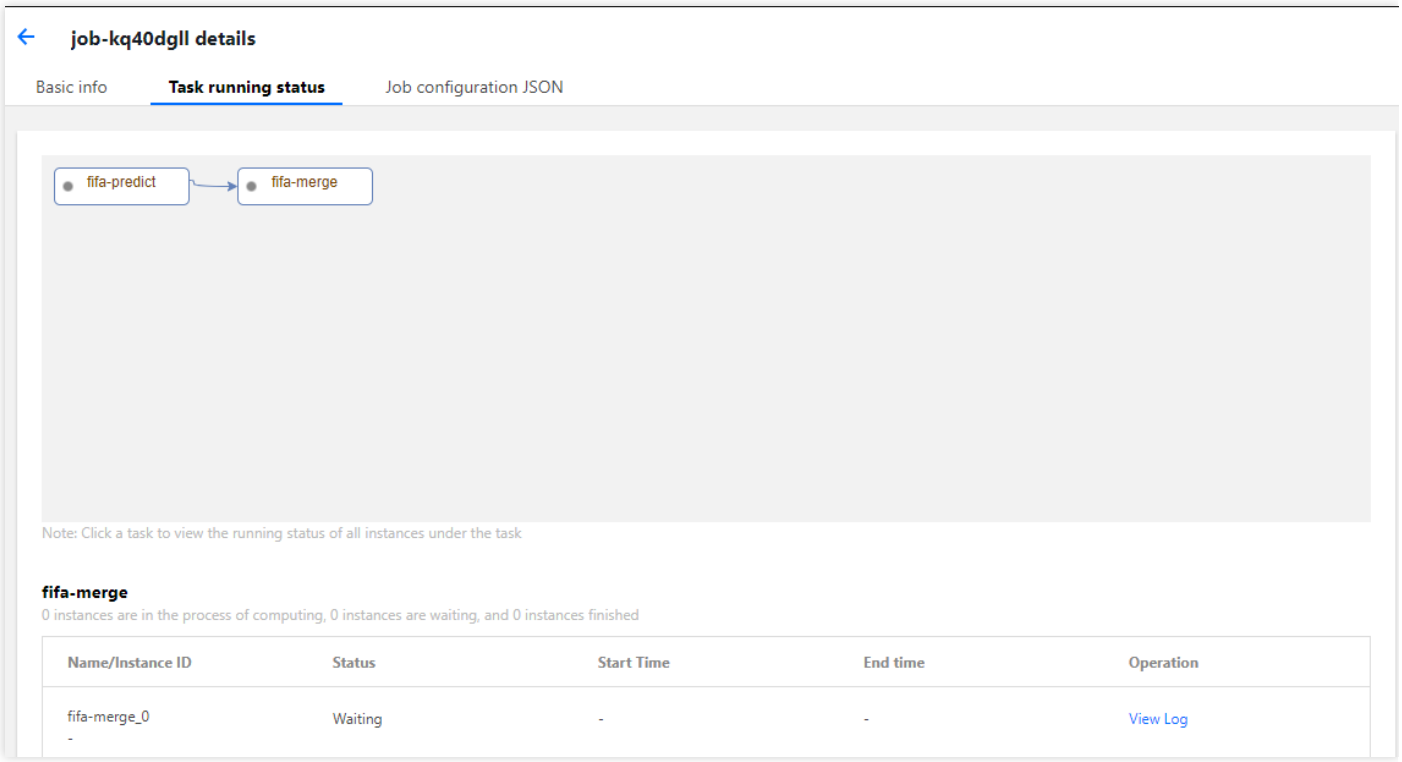
Step 5. Submit a job

1. Click **Job** on the left sidebar to enter the "Job" list page.
2. On the "Job" list page, select the target region at the top and click **Create**.
3. Enter the "New job" page and configure job information as shown below:
   - **Job name**: fifa.
   - **Priority**: default value.
   - **Description**: fifa 2018 model.
4. Select the **fifa-predict** and **fifa-merge** tasks on the left on the task flow page and drag them to the canvas on the right. Click the **fifa-predict** task anchor and drag it to the **fifa-merge** task.



5. Enable **Task information** on the right on the task flow page, confirm that the configuration is correct, and click **Complete**.

6. Query the job running information. For more information, please see Information Query.



7. Query the rendering result. For more information, please see Viewing Object Information.

Subsequent Operations

This document illustrates a simple machine learning job to demonstrate basic BatchCompute capabilities. You can continue to test the advanced capabilities of BatchCompute as instructed in the Console User Guide.

- **Various CVM configurations**: BatchCompute provides a variety of CVM configuration options. You can customize your own CVM configuration based on your business scenario.

- **Remote storage mapping**: BatchCompute optimizes storage access and simplifies access to remote storage services into operations in the local file system.
- **Concurrent multi-model training**: With BatchCompute, you can specify the number of concurrent instances and use environment variables to separate instances and read different training data for concurrent modeling.

Building Cluster Using Compute Environment

Last updated : 2019-10-18 14:47:32

Scenario

With the capabilities of BatchCompute (Batch), you can easily and efficiently maintain the Cloud Virtual Machine (CVM) cluster. The Batch computing environment corresponds to common cluster concepts. This document describes how to use the capabilities of the Batch computing environment to create or terminate an ultra cost-effective resource cluster.

Prerequisites

You can get prepared as instructed by Preparation.

Steps

Installing and Configuring TCCLI

> In the current computing environment, you can only call command lines. Install TCCLI by referring to the following steps:

1. Install TCCLI by referring to Preparation.
2. Run the following command to verify whether TCCLI is successfully installed:

```
tccli batch help
```

The returned result is as follows, indicating that TCCLI is successfully installed:

```
NAME
batch
DESCRIPTION
batch-2017-03-12
USEAGE
tccli batch <action> [--param...]
```

```
OPTIONS
help
show the tccli batch help info
--version
specify a batch api version
AVAILABLE ACTION
DescribeComputeEnv
Used to query details of the computing environment
CreateTaskTemplate
Used to create a task template
```

3. Configure TCCLI by referring to Preparation.

Creating a Computing Environment

You can acquire and modify the official example to create a Batch computing environment under a personal account. Learn each configuration item in the computing environment by referring to the following information.
You can also refer to the APIs related to the computing environment, for example, CreateComputeEnv.

The following example shows how to create a cluster with ten BS1.LARGE8 instances (Standard BatchCompute model, 4-core CPU and 8 GB memory) in Guangzhou Zone 2:

```
tccli batch CreateComputeEnv --version 2017-03-12 --ComputeEnv '{
"EnvName": "batch-env", // Computing environment name
"EnvDescription": "batch env demo", // Computing environment description
"EnvType": "MANAGED", // Computing environment type: MANAGED
"EnvData": { // Specific configuration (Refer to the CVM instance creation description.)
"InstanceType": "BS1.LARGE8", // CVM instance type in a computing environment
"ImageId": "img-m4q71qnf", // CVM image ID in a computing environment, which can be replaced by a
custom image ID
"LoginSettings": {
"Password": "B1[habcd" // CVM login password in a computing environment
},
"InternetAccessible": {
"PublicIpAssigned": "TRUE", // Whether the CVM requires a public IP address in a computing enviro
nment
"InternetMaxBandwidthOut": 10 // CVM bandwidth cap in a computing environment
},
"SystemDisk": {
"DiskType": "CLOUD_BASIC", // Type of a CVM disk in a computing environment (HDD cloud disk is us
ed now)
"DiskSize": 50 // Size of a CVM disk in a computing environment
}
},
```

```
"DesiredComputeNodeCount": 10 // Number of desired compute nodes
}'
--Placement'{
"Zone": "ap-guangzhou-2" // Availability zone (Guangzhou Zone 2 now, you can change it if necessa
ry)
}'
```

Sample Request

```
tccli batch CreateComputeEnv --version 2017-03-12 --ComputeEnv '{"EnvName":"batch-env","EnvDescri
ption":"batch env demo","EnvType":"MANAGED","EnvData":{"InstanceType":"BS1.LARGE8","ImageId":"img
-m4q71qnf","LoginSettings":{"Password":"B1[habcd"},"InternetAccessible":{"PublicIpAssigned":"TRU
E","InternetMaxBandwidthOut":50},"SystemDisk":{"DiskType":"CLOUD_BASIC","DiskSize":50}},"DesiredC
omputeNodeCount":1}' --Placement '{"Zone": "ap-guangzhou-2"}'
```

Response Example

In the following return values, `EnvId` indicates the unique ID of a Batch computing environment. The following will describe how to use Batch command line interface (CLI) to check the computing environment and instance information within it, and `EnvId` will be used. You need to record the returned value of `EnvId` .

```
{
"EnvId": "env-jlatqfkn",
"RequestId": "297ed003-7373-4950-9721-242d3d40b3ca"
}
```

You can view the created CVM in CVM Console or view and manage the CVM by using the CreateComputeEnv API.

Viewing the List of Computing Environments

You can use the Batch CLI to view the list of all created computing environments.

Sample Request

Run the following command to view the list of computing environments:

```
tccli batch DescribeComputeEnvs --version 2017-03-12
```

Response Example

The following result contains information about the computing environment to be queried (some information is omitted):

```
{
"TotalCount": 1,
"ComputeEnvSet": [
{
"EnvId": "env-jlatqfkn",
"ComputeNodeMetrics": {
...
},
"EnvType": "MANAGED",
"DesiredComputeNodeCount": 2,
"EnvName": "test compute env",
"Placement": {
...
},
"CreateTime": "2019-10-08T08:55:12Z"
}
],
"RequestId": "7a1f9338-0118-46bf-b59f-60ace9f154f5"
}
```


Viewing the Specified Computing Environment and Its Node List

Sample Request

Run the following command to view the specified computing environment and its node list:

```
tccli batch DescribeComputeEnv --version 2017-03-12 --EnvId env-jlatqfkn
```

Response Example

The following result contains the overall computing environment and details of each node (some information is omitted):


```
{
"EnvId": "env-jlatqfkn",
"ComputeNodeMetrics": {
...
},
"EnvType": "MANAGED",
"DesiredComputeNodeCount": 2,
"ComputeNodeSet": [
...
],
"RequestId": "407de39c-1c3d-489e-9a35-5257ae561e87",
"Placement": {
```

```
...
},
"EnvName": "test compute env",
"CreateTime": "2019-10-08T08:55:12Z"
}
```

Terminating a Computing Environment

Sample Request

Run the following command to terminate the computing environment. After you run the command, the computing environment automatically terminates all CVMs in the cluster.

```
tccli batch DeleteComputeEnv --version 2017-03-12 --EnvId env-jlatqfkn
```

Response Example

```
{
"RequestId": "029becda-2a4e-4989-aa77-6fbb5a873555"
}
```