

**批量计算
最佳实践
产品文档**



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

最佳实践

使用计算环境搭建集群

3ds Max 2018 渲染示例

深度学习示例

最佳实践

使用计算环境搭建集群

最近更新时间：2024-01-13 11:19:29

操作场景

使用批量计算（Batch）的计算环境能力，可以轻松高效的维护云服务器集群。Batch 的计算环境可以简单的对应常规的集群概念，本文介绍如何使用计算环境能力来快速创建/销毁一个超性价比资源集群。

前提条件

请参见 [开始前的准备](#) 中的步骤做好准备。

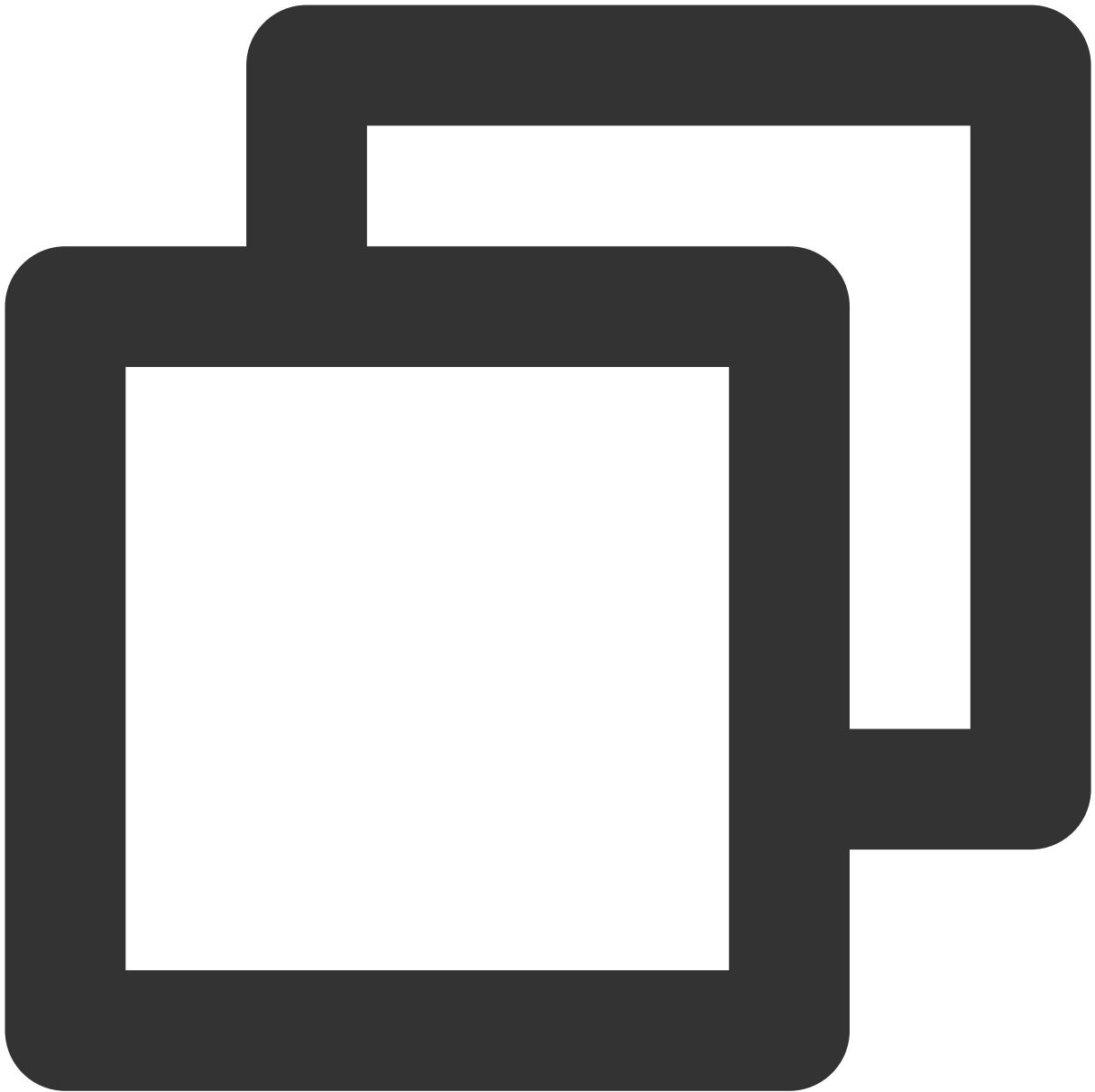
操作步骤

安装和配置命令行工具 TCCLI

说明：

目前计算环境仅支持命令行调用，请按照以下步骤安装命令行工具。

1. 请参见 [安装 TCCLI](#) 安装命令行工具。
2. 执行以下命令，验证 TCCLI 是否成功安装。



```
tccli batch help
```

返回结果如下，则成功安装。



```
NAME
    batch
DESCRIPTION
    batch-2017-03-12
USAGE
    tccli batch <action> [--param...]
OPTIONS
    help
    show the tccli batch help info
    --version
    specify a batch api version
```

AVAILABLE ACTION

DescribeComputeEnv
用于查询计算环境的详细信息

CreateTaskTemplate
用于创建任务模板

3. 请按照 [配置 TCCLI](#) 配置命令行工具。

创建计算环境

您可获取并修改官方提供的示例，作为个人账号下可执行的 Batch 计算环境。请参见以下内容了解计算环境各项配置的含义：

您也可查阅 [创建计算环境](#) 等计算环境相关接口。

以下示例在广州二区快速创建包含10台 BS1.LARGE8（批量计算通用型 CPU 4核 内存 8GB）类型集群。



```
tccli batch CreateComputeEnv --version 2017-03-12 --ComputeEnv '{
  "EnvName": "batch-env",           // 计算环境名称
  "EnvDescription": "batch env demo", // 计算环境描述
  "EnvType": "MANAGED",             // 计算环境类型, 托管型
  "EnvData": {                      // 具体配置 (可参照 CVM 创建实例说明)
    "InstanceType": "BS1.LARGE8",   // 计算环境内CVM 实例类型
    "ImageId": "img-m4q71qnf",      // 计算环境内CVM 镜像 ID (可替换成自定义镜像)
    "LoginSettings": {
      "Password": "B1[habcd"        // 计算环境内CVM 登录密码
    },
    "InternetAccessible": {
```



```
    "PublicIpAssigned": "TRUE", // 计算环境内CVM 是否需要公网IP
    "InternetMaxBandwidthOut": 10 // 计算环境内CVM 带宽上限
  },
  "SystemDisk": {
    "DiskType": "CLOUD_BASIC", // 计算环境内CVM 磁盘类型 (目前是普通云硬盘)
    "DiskSize": 50 // 计算环境内CVM 磁盘大小
  }
},
"DesiredComputeNodeCount": 10 // 计算节点期望个数
}'
--Placement '{
  "Zone": "ap-guangzhou-2" // 可用区 (当前广州二区可能需替换)
}'
```

请求示例



```
tccli batch CreateComputeEnv --version 2017-03-12 --ComputeEnv '{"EnvName":"batch-e
```

返回示例

返回值如下所示，其中 `EnvId` 为 Batch 计算环境的唯一标识。

后续将介绍如何通过 Batch 的命令行接口来查看计算环境以及计算环境内的实例信息，会使用到 `EnvId`，请记录返回的 `EnvId`。



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

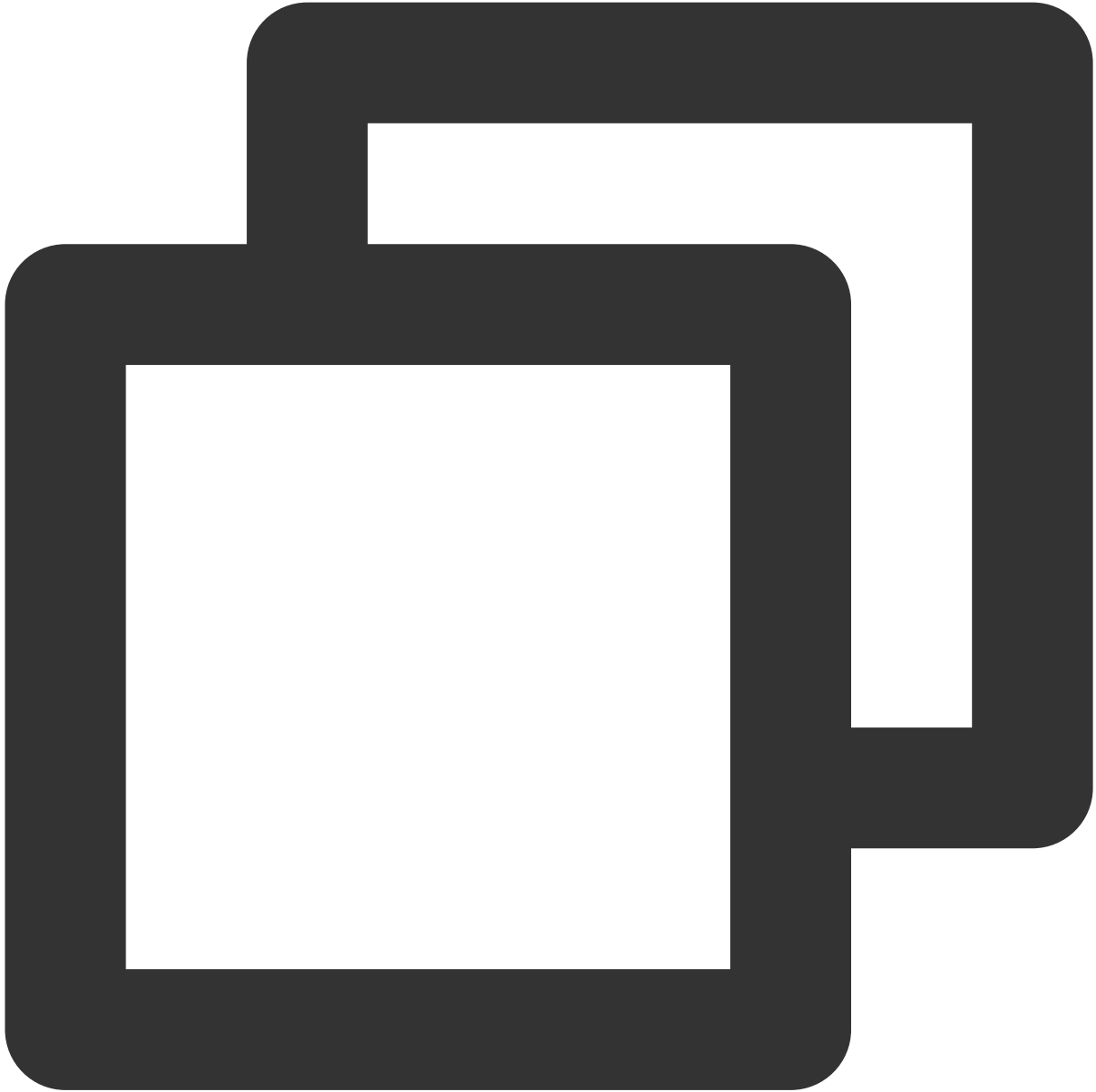
创建的主机可以通过 [云服务器控制台](#) 来查看，也可以通过 Batch 的 [计算环境接口](#) 来查看和管理。

查看计算环境列表

您可以通过 Batch 的命令行接口来查看创建的所有计算环境列表。

请求示例

执行以下命令，查看计算环境列表。



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

返回示例

返回结果如下，其中包含了所要查询的计算环境信息（部分已省略）。



```
{
  "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"  
}
```

查看指定计算环境以及包含的节点列表

请求示例

执行以下命令，查看指定计算环境以及包含的节点列表。



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

返回示例

返回结果如下，其中包含了计算环境整体，以及每个节点的详细信息（部分已省略）。



```
{
  "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"  
}
```

销毁计算环境

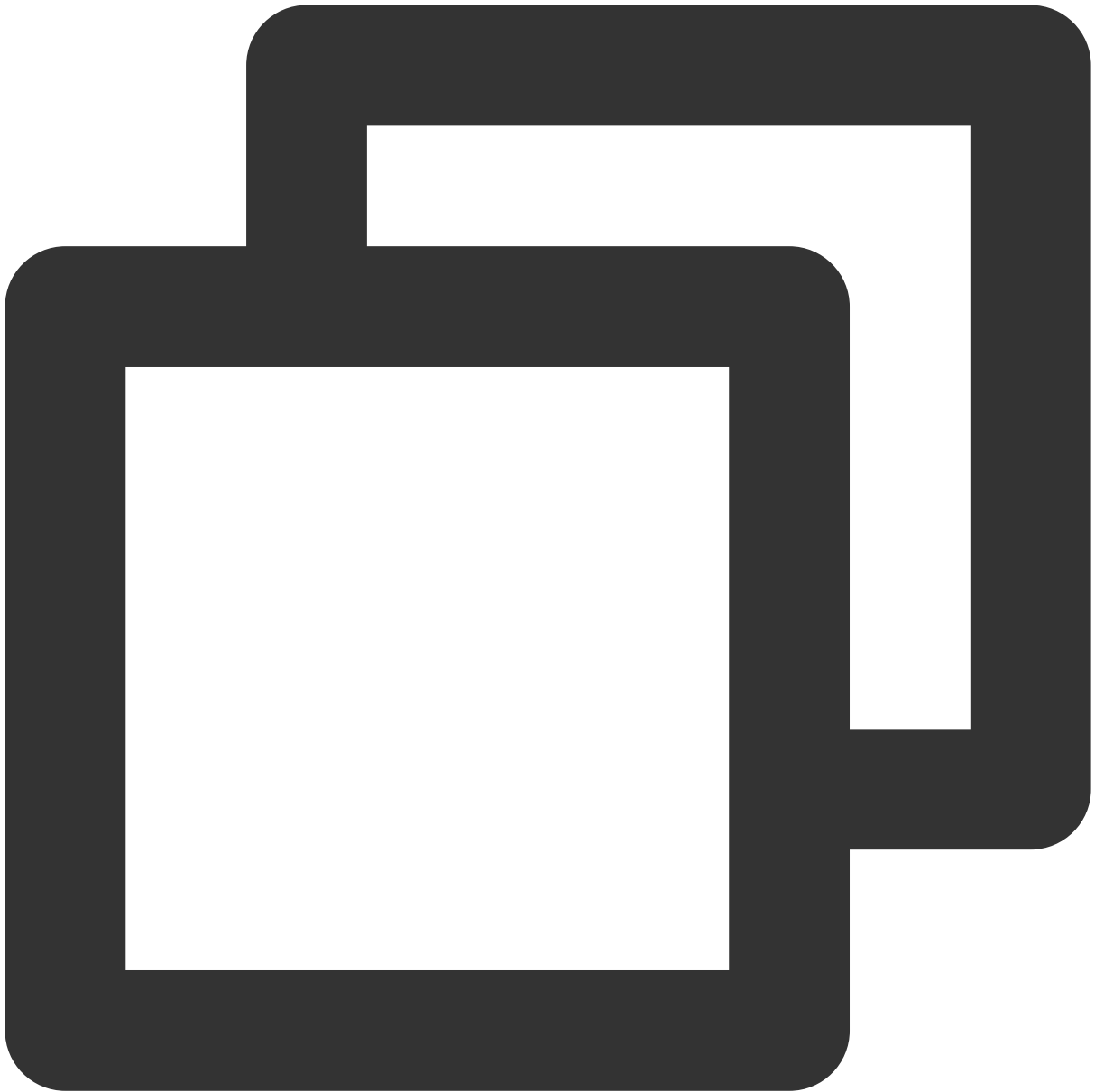
请求示例

执行以下命令，销毁计算环境，调用后计算环境会自动销毁集群内所有的云服务器。



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

返回示例



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

3ds Max 2018 渲染示例

最近更新时间：2024-01-13 11:19:28

快速开始

本文介绍如何使用批量计算控制台提交一个作业，完成 3ds Max 2018 图片渲染，并导出渲染图片。具体操作步骤如下：

步骤1：制作自定义镜像

1. 自定义镜像流程，参见 [Windows自定义镜像](#)。
2. 3ds Max 2018 安装流程，参见 [官方主页](#)。

注意：

请暂时关闭 Windows 防火墙，避免阻止软件下载。

参见 [显卡选型指南](#) 选择合适的显卡类型，避免显卡初始化失败。无特殊原因，推荐选择“Nitrous Software”。

步骤2：准备渲染文件

渲染素材的主流存储方式有两种：[对象存储](#) 和 [文件存储](#)。通过配置挂载参数，Batch 会在渲染作业运行前挂载对象存储或文件存储到本地，渲染器可以像访问本地文件一样访问对象存储或文件存储。

渲染素材较小时，推荐将全部素材压缩成 gzip 包，上传至对象存储。详情请参见 [上传对象](#)。

渲染素材较大时，推荐存放在文件存储上。

步骤3：创建任务模板

1. 登录批量计算控制台，选择左侧导航栏中的 [任务模板](#)。
2. 在 [任务模板](#) 页面上方选择目标地域后，单击 [新建](#)。
3. 单击 [新建](#)，进入 [新建任务模板](#) 页面，参考以下信息进行创建。如下图所示：

New task template

1 Basic Configuration > 2 Program running configuration > 3 Storage mapping

Basic Information

Name: rendering

Description: 3ds Max 2018 Demo

Compute environment type: Existing compute environment | **Auto compute environment**

Resource configuration: S1.LARGE8 (4-core, 8 GB) | [CVM Detailed Configuration](#)

System disk (50 GB) Bandwidth (No public network bandwidth), password (system-generated)

Image: Public Images | Windows Server 2012 R2 DataCenter 64bitEN

You must select the images that have installed and configured Cloud-init.

Resource quantity: - 1 +

Timeout threshold: 259200 sec

Number of retry attempts: 0

Tag configuration

Tag key	Tag value	Operation
Please select	Please select	×

[Add](#)

Next

名称：自定义，例如 rendering。

描述：自定义，例如 3ds Max 2018 Demo。

计算环境类型：请按需选择，本文选择**自动计算环境**。

资源配置：S1.LARGE8（4核8G）。

镜像：自定义镜像标识符，请按需选择。

资源数量：并发渲染数，例如1台。

超时时间、重试次数：保持默认值。

4. 单击**下一步**，配置程序信息。如下图所示：

The screenshot shows the 'Program running configuration' step in the Tencent Cloud console. The interface is divided into two main sections: 'Program configuration' and 'Command line'. In the 'Program configuration' section, there are four rows of configuration options: 'Execution method' (set to 'Package'), 'Package address' (with a 'Check' button), 'Stdout log' (with a 'Check' button), and 'Stderr log' (with a 'Check' button'). The 'Command line' section contains a text input field with the command '3dsmaxcmd Demo.max -outp'. At the bottom of the form, there are two buttons: 'Previous' and 'Next'.

执行方式：PACKAGE。

程序包地址：以对象存储举例，`cos://barrygz-125xxxxxx4.cos.ap-guangzhou.myqcloud.com/render/max.tar.gz`。

Stdout日志：格式参见 [COS](#)、[CFS](#) 路径填写。

Stderr日志：同 Stdout 日志。

命令行：`3dsmaxcmd Demo.max -outputName:c:\\\\render\\\\image.jpg`。

5. 单击**下一步**配置存储映射。如下图所示：

Basic Configuration > Program running configuration > **3** Storage mapping configuration > 4

Input path mapping

Copy the data you want to process from COS/CFS to the local disk of your CVM

COS/CFS path	Local path

Activate

Output path mapping

Copy the computing results from the local disk of your CVM to the COS/CFS

Local path	COS/CFS path
C:\\render\\	cos://barrygz-1251783334.cos.ap-guangzhou.myqcl

Activate

Previous Next

输出路径映射-本地路径： C:\\\\render\\\\ 。

输出路径映射-COS CFS路径：格式参考 [COS](#)、[CFS](#) 路径填写。

6. 单击**下一步**预览任务 JSON 文件。

7. 确认无误后，单击**保存**即可。

步骤4：提交作业

1. 单击左侧导航栏中的**作业**，进入**作业**列表页面。

2. 在页面上方选择目标地域后，单击**新建**。

3. 在**新建作业**窗口中，配置作业基本信息。如下图所示：

Job name	<input type="text" value="max"/>
Priority	<input type="text" value="0"/>
	It should range from 0 to 100. A higher value means a higher priority
Description	<input type="text" value="3ds Max 2018 Demo"/>

作业名称：max。

优先级：默认值。

描述：3ds Max 2018 Demo。

4. 选中任务流页面左侧 **rendering** 任务，移动鼠标将任务放置到右侧画布中。

Task flow
You can set dependencies between different tasks here.
Click to select the task on the left, and move the mouse cursor to place the task on the canvas on the right. Drag delete the element.

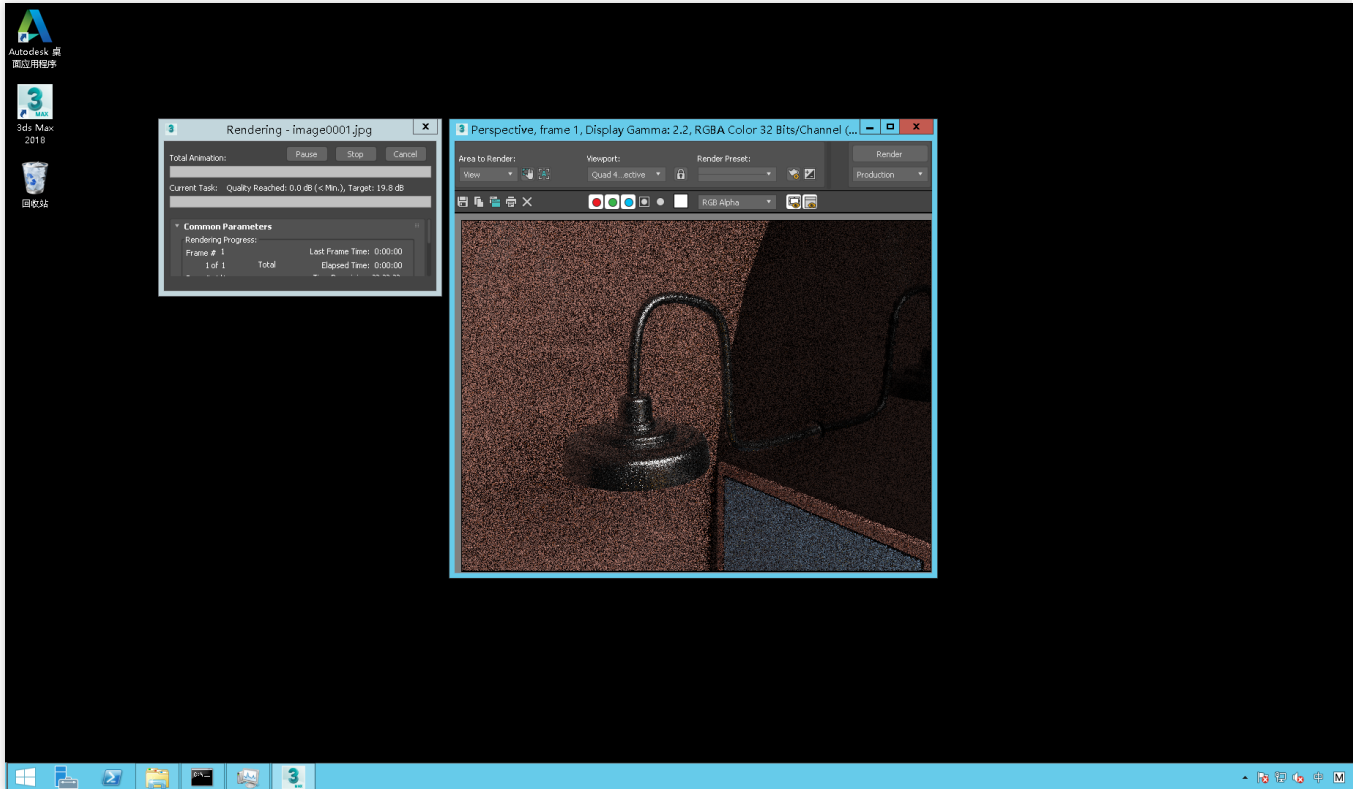
Task Template

- rendering
- post-task
- pre-task2
- pre-task1
- hello

rendering

Completed Cancel

5. 打开任务流右侧 **任务详情**，确认配置无误后，单击**完成**。
6. 查询作业运行信息，请参见 [查询信息](#)。
7. 渲染过程演示。



8. 渲染结果查询，请参见 [查看对象信息](#)。

下一步操作

本文列举了一个简单的渲染示例，它是一个单实例的作业，仅仅是向用户展示最基本的能力，您可以根据控制台使用指南继续测试 Batch 更高阶的能力。

丰富的云服务器配置：Batch 提供了丰富的云服务器 CVM 配置项，您可以根据业务场景自定义 CVM 配置。

远程存储映射：Batch 在存储访问上进行优化，将对远程存储服务的访问简化为对本地文件系统操作。

并行渲染多张图片：Batch 支持指定并发渲染数，通过 [环境变量](#) 区分不同的渲染实例，每个渲染实例读取不同的渲染素材，实现并行渲染。

深度学习示例

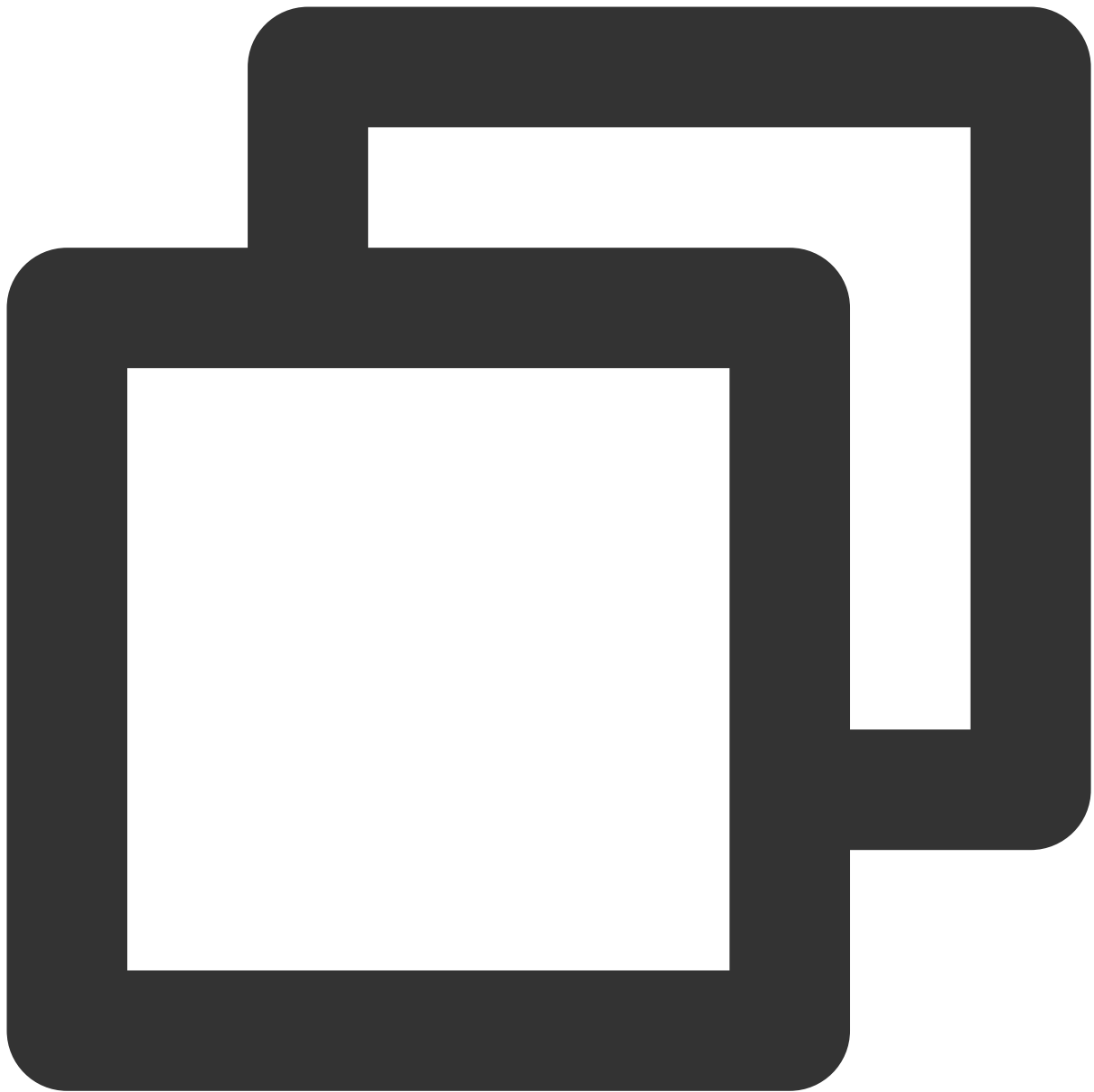
最近更新时间：2024-01-13 11:19:29

快速开始

本文介绍一个基于 `scikit-learn` 机器学习库，编写一个多层感知器（MLP，Multilayer Perceptron）BP 算法的深度学习示例。通过对历史国际足球比赛、球队排名、球员体能技术指标以及 FIFA 2018 小组赛结果建模，预测两只球队的胜负平概率。具体操作步骤如下。

步骤1：制作自定义镜像

1. 制作步骤参见 [创建自定义镜像](#) 文档。
2. 安装依赖包，以 CentOS 7.2 64 bit 为例：



```
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
```

步骤2：下载程序包

单击 [下载程序压缩包](#)，下载完成后上传压缩包至 [对象存储](#)。通过指定程序包的对象存储地址，Batch 会在作业运行前下载压缩包到云服务器，自动解压后执行。

步骤3：创建“fifa-predict”任务模板

1. 登录批量计算控制台，选择左侧导航栏中的 [任务模板](#)。
2. 在 [任务模板](#) 页面上方选择目标地域后，单击 [新建](#)。
3. 单击 [新建](#)，进入 [新建任务模板](#) 页面，参考以下信息进行创建。如下图所示：

New task template

1 Basic Configuration > 2 Program running configuration > 3 Storage mapping configuration > 4 JSON p

Basic Information

Name:

Description:

Compute environment type①: Existing compute environment Auto compute environment

Resource configuration: S2.SMALL1 (1-core, 1 GB) [CVM Detailed Configuration](#)

System disk (50 GB) Bandwidth (No public network bandwidth), password (system-generated)

Image: Custom Images Please select

You must select the images that have installed and configured Cloud-init.①

Resource quantity:

Timeout threshold①: sec

Number of retry attempts①:

Tag configuration

Tag key	Tag value	Operation
<input type="text" value="Please select"/>	<input type="text" value="Please select"/>	<input type="text" value="x"/>

[Add](#)

名称：fifa-predict。

描述：数据训练与预测。

计算环境类型：请按需选择，本文选择 [自动计算环境](#)。

资源配置：S2.SMALL1（1核1G），公网带宽按量收费。

镜像：自定义镜像标识符，选择 [步骤1](#) 中已创建的镜像。

资源数量：并发渲染数，例如3台，并行训练3个神经网络模型。

超时时间、重试次数：请保持默认值。

4. 单击**下一步**，配置程序信息。如下图所示：

The screenshot shows a web interface for creating a new task template. It has three steps: 1. Basic Configuration, 2. Program running configuration (current), and 3. Storage mapping configuration. Under 'Program configuration', there are four input fields: 'Execution method' (dropdown menu with 'Package' selected), 'Package address' (text input with 'cos://barryg...' and a 'Check' button), 'Stdout log' (text input with 'cos://barryg...' and a 'Check' button), and 'Stderr log' (text input with 'cos://ba...' and a 'Check' button). Below this is a 'Command line' section with a text input containing 'echo hello world'. At the bottom are 'Back' and 'Next' buttons.

执行方式：PACKAGE。

程序包地址：以对象存储举例，`cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz`。

Stdout 日志：格式参见 [COS](#)、[CFS](#) 路径填写。

Stderr 日志：同 Stdout 日志。

命令行：`python predict.py "Japan" "Senegal"`。

球队列表：'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. 跳过存储映射配置步骤，单击**下一步**。

6. 预览任务 JSON 文件，确认无误后，单击**保存**。

步骤4：创建“fifa-merge”任务模板

1. 登录批量计算控制台，选择左侧导航栏中的 [任务模板](#)。

2. 在“任务模板”页面上方选择目标地域后，单击**新建**。

3. 单击**新建**，进入**新建任务模板**页面，参考以下信息进行创建。如下图所示：

New task template

1 Basic Configuration > 2 Program running configuration > 3 Storage mapping configuration > 4 JSON preview

Basic Information

Name: fifa-merge

Description: Aggregation of prediction data

Compute environment type: Existing compute environment | **Auto compute environment**

Resource configuration: S2.SMALL1 (1-core, 1 GB) | CVM Detailed Configuration

Image: Custom Images | kafutest

Resource quantity: 1

Timeout threshold: 259200 sec

Number of retry attempts: 0

Tag configuration:

Tag key	Tag value	Operation
Please select	Please select	×

Add

Next

名称：fifa-merge。

描述：预测数据汇总。

计算环境类型：请按需选择，本文选择**自动计算环境**。

资源配置：S2.SMALL1（1核1G），公网带宽按量收费。

镜像：自定义镜像标识符，选择**步骤1**中已创建的镜像。

资源数量：1台。

超时时间、重试次数：请保持默认值。

4. 单击**下一步**，配置程序信息。如下图所示：

New task template

Basic Configuration > **2** Program running configuration > 3 Storage mapping configuration >

Program configuration

Execution method: Package

Package address: [Check](#)

Stdout log: [Check](#)

Stderr log: [Check](#)

Command line

[Back](#) [Next](#)

执行方式：PACKAGE。

程序包地址：以对象存储举例， `cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz`。

Stdout 日志：格式参见 [COS](#)、[CFS](#) 路径填写。

Stderr 日志：同 Stdout 日志。

命令行： `python merge.py /data`。

5. 单击**下一步**，配置存储映射。如下图所示：

Basic Configuration > Program running configuration > 3 Storage mapping configuration

Input path mapping

Copy the data you want to process from COS/CFS to the local disk of your CVM

COS/CFS path	Local path
cos://[redacted]uangzhou.myqclo Check	/data/

[Activate](#)

Output path mapping

Copy the computing results from the local disk of your CVM to the COS/CFS

Local path	COS/CFS path
------------	--------------

[Activate](#)

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

输入路径映射 - COS/CFS路径：填写“fifa-predict”模板 Stdout 日志路径。

输入路径映射 - 本地路径： /data/。

6. 预览任务 JSON 文件，确认无误后，单击**保存**。

步骤5：提交作业

1. 单击左侧导航栏中的**作业**，进入**作业列表**页面。
2. 在**作业列表**页面上方选择目标地域后，单击**新建**。
3. 进入**新建作业**页面，参考以下信息配置作业信息。

作业名称：fifa。

优先级：默认值。

描述：fifa 2018 model。

4. 选中任务流页面左侧 **fifa-predict** 和 **fifa-merge** 任务，移动鼠标将任务放置到右侧画布中。单击 **fifa-predict** 任务拖拽箭头到 **fifa-merge** 任务。

Task flow
You can set dependencies between different tasks here.

Click to select the task on the left, and move the mouse cursor to place the task on the canvas on the right. Drag the anchor to connect the task to other tasks. Click on the selected element in the canvas and press "Delete" to delete the element. info

Task Template

- fifa-merge
- fifa-predict
- rendering
- post-task
- pre-task2
- pre-task1
- hello

```

graph LR
    A[fifa-predict] --> B[fifa-merge]
            
```

fifa-merge

Basic info

Name: fifa-merge

Resource configuration: S2.SMALL1 (1-core, 1 GB)
System disk (50 GB), data disk (0GB) ⓘ

Concurrent instances: 1

Command line: python merge.py /data


ID: task-tmpl-ds6oweba

Creation Time: 2018-11-30 17:33:49

5. 打开任务流右侧**任务详情**，确认配置无误后，单击**完成**。
6. 查询作业运行信息，请参见 [查询信息](#)。

← job-kq40dgll details

Basic info **Task running status** Job configuration JSON



Note: Click a task to view the running status of all instances under the task

fifa-merge
0 instances are in the process of computing, 0 instances are waiting, and 0 instances finished

Name/Instance ID	Status	Start Time	End time
fifa-merge_0	Waiting	-	-

7. 渲染结果查询，请参见 [查看对象信息](#)。

View Log

```
The winner of Senegal and Japan is Senegal
Probability of Senegal winning is 0.466
Probability of draw is 0.257
Probability of Japn winning is 0.277
```

StdOutput log StdErr log

下一步操作

本文列举了一个简单的机器学习示例，仅仅是向用户展示最基本的能力，您可以根据控制台使用指南继续测试 Batch 更高阶的能力。

丰富的云服务器配置：Batch 提供了丰富的云服务器 CVM 配置项，您可以根据业务场景自定义 CVM 配置。

远程存储映射：Batch 在存储访问上进行优化，将对远程存储服务的访问简化为对本地文件系统操作。

并行训练多个模型：Batch 支持指定并发数，通过 [环境变量](#) 区分不同的并发实例，每个实例读取不同的训练数据，实现并行建模。