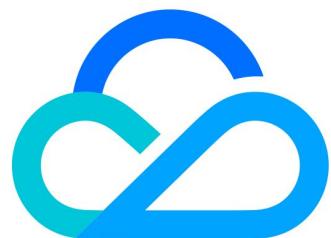


批量计算

最佳实践

产品文档



腾讯云

【版权声明】

©2013-2018 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

3ds Max 2018 渲染示例

深度学习示例

使用计算环境搭建集群

最佳实践

3ds Max 2018 渲染示例

最近更新时间：2018-06-14 16:40:47

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：

一. 制作自定义镜像

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

注意：

- 请暂时关闭 Windows 防火墙，避免阻止软件下载。
- 参考 [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”。

二. 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。可参考 [上传对象](#)。
- 渲染素材较大时，推荐存放在文件存储上。

三. 创建任务模板

1. 登录 [批量计算控制台](#)，单击左侧导航栏【任务模板】选项，选择目标地域后，单击【新建】按钮。

2. 配置基本信息。示例如下：

基本信息

名称	rendering	
描述	3ds Max 2018 Demo	
资源配置	S1.LARGE8 (4核8GB)	云主机详细配置
系统盘 (50GB)、带宽 (按流量计算10Mbps)、密码 (自定义密码)		
资源数量	- 1 +	台
超时时间①	259200	秒
重试次数①	0	▼
镜像	img-i64lx84h	
下一步		

- 名称：rendering
- 描述：3ds Max 2018 Demo
- 资源配置：S1.LARGE8 (4核8G)
- 资源数量：并发渲染数，比如 1 台
- 超时时间：默认值
- 重试次数：默认值
- 镜像：自定义镜像标识符，比如 img-i64lx84h

3. 配置程序信息。示例如下：

The screenshot shows the 'Program Configuration' section of a configuration wizard. It includes fields for execution mode (set to 'PACKAGE'), program package address (cos://barrygz-1251783334.cos.ap-guangzhou.myqcloud.com/render/max.tar.gz), and log paths for Stdout and Stderr, both set to the same address. A 'Check' button is visible next to each entry. Below this, the 'Command Line' section contains the command '3dsmaxcmd Demo.max -outputName:c:\\render\\image.jpg'. At the bottom are 'Previous Step' and 'Next Step' buttons, with 'Next Step' being highlighted.

- 执行方式：PACKAGE
- 程序包地址：以对象存储举例， cos://barrygz-1251783334.cos.ap-guangzhou.myqcloud.com/render/max.tar.gz
- Stdout日志：格式参考 [COS、CFS 路径填写](#)
- Stderr日志：同 Stdout 日志
- 命令行： 3dsmaxcmd Demo.max -outputName:c:\\render\\image.jpg

4. 配置存储映射。

输入路径映射

将您要处理的数据，从COS / CFS存储复制到云主机的本地磁盘中

COS / CFS路径	本地路径

[添加](#)

输出路径映射

将计算返回的结果，从云主机本地磁盘复制到COS / CFS存储中

本地路径	COS / CFS路径	校验通过
C:\render\	cos://barrygz-1251783334.cos.ap-guangzhou.myqcloud.cn	校验

[添加](#)

[上一步](#) [下一步](#)

- 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](#)

5. 预览任务 JSON 文件，确认无误后，单击【保存】按钮。

四. 提交作业

1. 单击左侧导航栏【作业】选项，选择目标地域后，单击【新建】按钮。

2. 配置作业基本信息。示例如下：

作业名称: max

优先级: 0

范围 0-100, 数值越大, 优先级越高

描述: 3ds Max 2018 Demo

- 作业名称 : max
- 优先级 : 默认值
- 描述 : 3ds Max 2018 Demo

3. 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。

4. 打开任务流右侧 任务详情, 确认配置无误后, 单击【完成】按钮。

任务流
您可以在这里设置不同任务之间的依赖关系。
点击选中左侧任务, 移动鼠标将任务放置到右侧画布中。拖拽锚点建立连接。点击选中画布中的元素, 按Delete键可删除元素

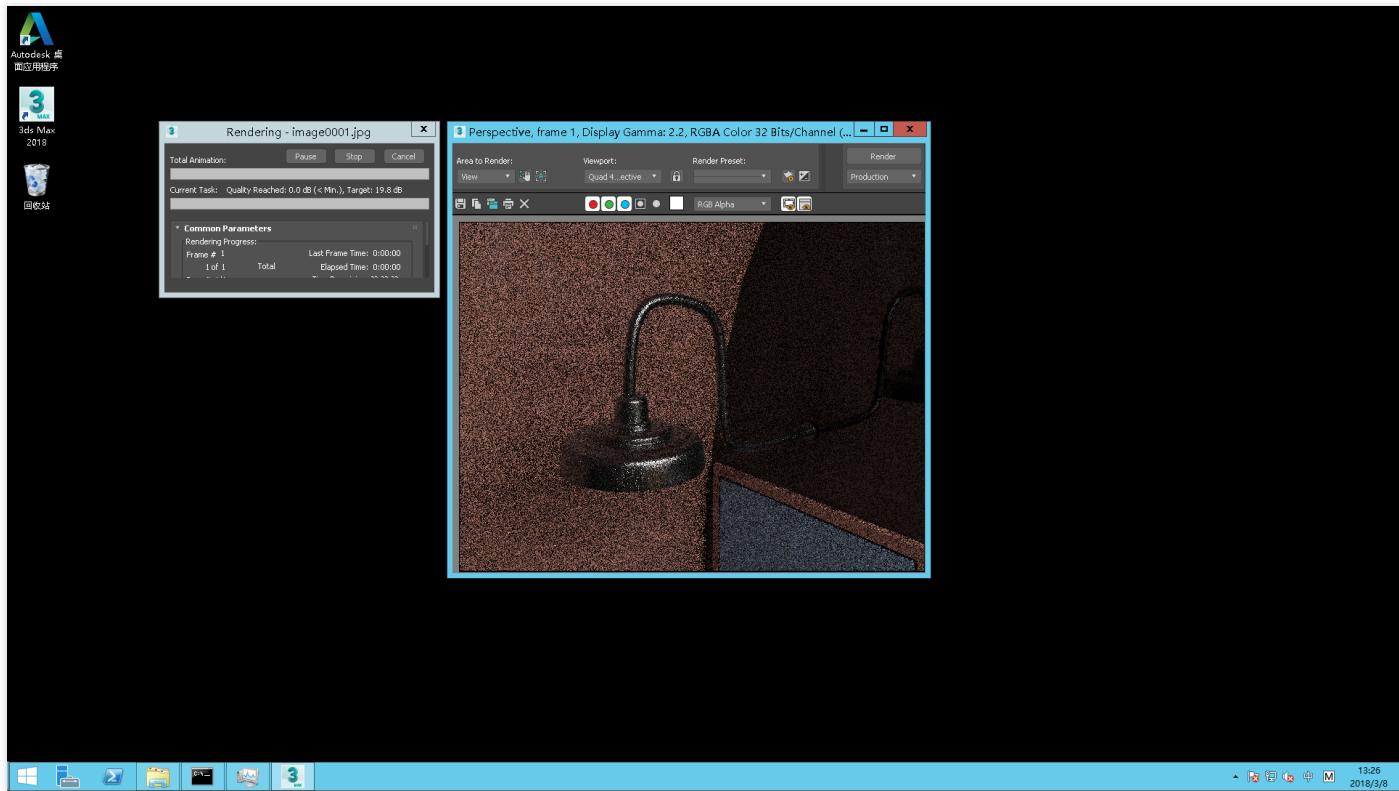
任务信息

任务模板	rendering
rendering	基本信息
A	名称: rendering
abc	资源配置: S1.LARGE8 (4核8GB)
s	系统盘 (50GB)、数据盘 (0GB)
test2018	并发实例数: 1
sssss	命令行: 3dsmaxcmd Demo.max -outputName:c:\render\image.jpg
ssss	ID: task-tmpl-pq24rq1e
tt111	创建时间: 2018-03-08 11:27:43
jitest	
barrytpl	
A	

完成 取消

5. 查询作业运行信息, 参考 [查询信息](#)。

6. 渲染过程演示。



7. 渲染结果查询，参考 [查看对象信息](#)。

下一步可以干什么？

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：2018-06-28 15:48:18

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：

一. 制作自定义镜像

1. 制作步骤参考 [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
```

二. 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。

三. 创建“fifa-predict”任务模板

1. 登录 [批量计算控制台](#)，单击左侧导航栏【任务模板】选项，选择目标地域后，单击【新建】按钮。

2. 配置基本信息。示例如下：

基本配置 程序运行配置 存储映射配置 JSON预览

基本信息

名称	fifa-predict
ID	task-tmpl-9r392atu
创建时间	2018-06-25 19:32:34
描述	数据训练与预测
资源配置	S2.SMALL1 (1核1GB) 系统盘 (50GB)、数据盘 (0GB)、带宽 (按流量计算1Mbps)、密码 (系统自动生成)
资源数量	- 3 + 台
超时时间①	259200 秒
重试次数①	0
镜像	img-2id9heh1

保存 **取消**

- 名称 : fifa-predict ;
- 描述 : 数据训练与预测 ;
- 资源配置 : S2.SMALL1 (1核1G) , 公网带宽按量收费 ;
- 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 : 默认值 ;
- 重试次数 : 默认值 ;
- 镜像 : 自定义镜像标识符 , 例如 img-i64lx84h.

3. 配置程序信息。示例如下：

基本配置 **程序运行配置** 存储映射配置 JSON预览

程序配置

执行方式 **Package**

程序包地址 **cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz** 校验

Stdout日志 **cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.log** 校验

Stderr日志 **cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.log** 校验

命令行

python predict.py "Japan" "Senegal"

保存 取消

- 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：格式参考 [COS、CFS 路径填写](#)；
- Stderr 日志：同 Stdout 日志；
- 命令行：**python predict.py "Japan" "Senegal"**。

4. 球队列表：'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. 配置存储映射（略过）。

6. 预览任务 JSON 文件，确认无误后，单击【保存】按钮。

四. 创建“fifa-merge”任务模板

1. 登录 [批量计算控制台](#)，单击左侧导航栏【任务模板】选项，选择目标地域后，单击【新建】按钮。

2. 配置基本信息。示例如下：

基本配置 程序运行配置 存储映射配置 JSON预览

基本信息

名称

ID

创建时间

描述

资源配置 云主机详细配置
系统盘 (50GB)、数据盘 (0GB)、带宽 (按流量计算1Mbps)、密码 (系统自动生成)

资源数量 台

超时时间 秒

重试次数

镜像

- 名称：fifa-merge；
- 描述：预测数据汇总；
- 资源配置：S2.SMALL1 (1核1G)，公网带宽按量收费；
- 资源数量：1 台；
- 超时时间：默认值；
- 重试次数：默认值；
- 镜像：自定义镜像标识符，例如 img-i64lx84h。

3. 配置程序信息。示例如下：

基本配置 程序运行配置 存储映射配置 JSON预览

程序配置

执行方式 Package

程序包地址 cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.tar.gz 校验

Stdout日志 cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.log 校验

Stderr日志 cos://barrygz-1251783334.cosgz.myqcloud.com/fifa/fifa.2018.err 校验

命令行

python2.7 merge.py /data

保存 取消

- 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：格式参考 [COS、CFS 路径填写](#)；
- Stderr 日志：同 Stdout 日志；
- 命令行： python merge.py /data 。

4. 配置存储映射。

基本配置 程序运行配置 存储映射配置 JSON预览

输入路径映射

将您要处理的数据，从COS / CFS存储复制到云主机的本地磁盘中

COS / CFS路径	本地路径
cos://.cos.ap-guangzhou.myqcloud.com	/data/ 校验

[添加](#)

输出路径映射

将计算返回的结果，从云主机本地磁盘复制到COS / CFS存储中

本地路径	COS / CFS路径
添加	添加

保存 取消

- 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。

5. 预览任务 JSON 文件，确认无误后，单击【保存】按钮。

五. 提交作业

1. 单击左侧导航栏【作业】选项，选择目标地域后，单击【新建】按钮。
2. 配置作业基本信息。示例如下：
 - 作业名称：fifa；
 - 优先级：默认值；
 - 描述：fifa 2018 model。
3. 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。

任务流

您可以在这里设置不同任务之间的依赖关系。

点击选中左侧任务，移动鼠标将任务放置到右侧画布中。拖拽锚点建立连接。点击选中画布中的元素，按Delete键可删除元素

任务信息

任务模板

- fifa-merge
- fifa-predict
- claudetest
- rendering

fifa-predict → fifa-merge

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

fifa-merge

基本信息

名称:

资源配置:

系统盘 (50GB)、数据盘 (0GB) (i)

并发实例数:

命令行:

ID: task-tmpl-3dytbg6c

创建时间: 2018-06-25 19:34:51

完成取消

4. 打开任务流右侧 **任务详情**，确认配置无误后，单击【完成】按钮。

5. 查询作业运行信息，参考 [查询信息](#)。

job-gjwxed7r 详情

基本信息 任务运行情况 [作业配置JSON](#)

提示：点击任务可查看任务下所有实例的运行状态

fifa-merge
0个实例正在计算，0个等待中，1个已完成

名称/实例ID	状态	开始时间	结束时间	操作
fifa-merge_0	已完成	2018-06-26 12:02:37	2018-06-26 12:04:23	查看日志

6. 渲染结果查询，参考 [查看对象信息](#)。

查看日志 X

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

StdOutput 日志 StdErr 日志

[关闭](#)

下一步可以干什么？

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。

使用计算环境搭建集群

最近更新时间：2018-06-25 11:38:47

如何快速创建集群？

使用批量计算的计算环境能力，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，下面的例子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，目前计算环境仅支持命令行调用，请先参照开始准备安装命令行工具。

开始前的准备

开始前请根据文档 [开始前的准备](#) 里的检查清单做好准备，同时本例将使用到命令行工具（CLI），用户需要先安装和配置命令行工具。

安装和配置命令行工具

配置命令行工具请查看 [配置命令行工具](#)，安装完后检查下安装成功。

```
qcloudcli batch help

CreateComputeEnv |DescribeJobSubmitInfo
CreateTaskTemplate |DescribeJobs
DeleteComputeEnv |DescribeTask
DeleteJob |DescribeTaskTemplates
DeleteTaskTemplates |ModifyTaskTemplate
DescribeAvailableCvmInstanceTypes |SubmitJob
DescribeComputeEnv |TerminateJob
DescribeComputeEnvs |TerminateTaskInstance
DescribeJob
```

创建计算环境

您可以从官方提供的例子来修改出一个在您的账号下可执行的 Batch 计算环境，在这之前先看一下计算环境配置各项代表什么含义。

您也可以参考 [创建计算环境](#) 等计算环境相关接口。

下面的例子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

```
qcloudcli 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

```
qcloudcli batch CreateComputeEnv --Version 2017-03-12 --ComputeEnv '{"EnvName":"batch-env","EnvDescription":"batch env demo","EnvType":"MANAGED","EnvData":{"InstanceType":"BS1.LARGE8","ImageId":"img-m4q71qnf","LoginSettings":{"Password":"B1[habcd"},"InternetAccessible":{"PublicIpAssigned":"TRUE","InternetMaxBandwidthOut":50}, "SystemDisk":{"DiskType":"CLOUD_BASIC","DiskSize":50}}, "DesiredComputeNodeCount":1}' --Placement '{"Zone": "ap-guangzhou-2"}'
```

返回示例

返回值，其中 EnvId 为 Batch 计算环境的唯一标识。

```
{  
    "Response": {  
        "EnvId": "env-c96rwhnf",  
        "RequestId": "bead16d4-b33b-47b5-9b86-6a02b4bed1b2"  
    }  
}
```

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，后面将介绍如何通过 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，可以记录返回的 EnvId。

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

```
qcloudcli batch DescribeComputeEnvs --Version 2017-03-12
```

返回示例

```
{
  "Response": {
    "TotalCount": 1,
    "ComputeEnvSet": [
      {
        "EnvId": "env-c96rwhnf",
        "Placement": {
          "Zone": "ap-guangzhou-2"
        },
        "EnvType": "MANAGED",
        "EnvName": "test compute env",
        "ComputeNodeMetrics": {
          "CreatedCount": 0,
          "DeletingCount": 0,
          "CreationFailedCount": 0,
          "SubmittedCount": 0,
          "CreatingCount": 0,
          "AbnormalCount": 0,
          "RunningCount": 2
        },
        "CreateTime": "2017-11-27T07:10:02Z"
      }
    ],
    "RequestId": "bac76f1c-06cd-4ef4-82a9-f230fa5a1992"
  }
}
```

刚才创建的计算环境会包含在返回结果里，同时也包含计算环境

查看指定计算环境以及包含的节点列表

请求示例

```
qcloudcli batch DescribeComputeEnv --Version 2017-03-12 --EnvId env-c96rwhnf
```

返回示例

```
{  
    "Response": {  
        "EnvId": "env-c96rwhnf",  
        "Placement": {  
            "Zone": "ap-guangzhou-2"  
        },  
        "EnvType": "MANAGED",  
        "EnvName": "test compute env",  
        "RequestId": "12dc7dba-f33b-4d5a-8cd6-ebd1df17ebf7",  
        "ComputeNodeMetrics": {  
            "CreatedCount": 0,  
            "DeletingCount": 0,  
            "CreationFailedCount": 0,  
            "SubmittedCount": 0,  
            "CreatingCount": 0,  
            "AbnormalCount": 0,  
            "RunningCount": 2  
        },  
        "ComputeNodeSet": [  
            {  
                "ComputeNodeId": "node-838udz1w",  
                "ComputeNodeState": "RUNNING",  
                "Mem": 2,  
                "ResourceCreatedTime": "2017-11-27T07:10:46Z",  
                "ComputeNodeInstanceId": "ins-q09nyg5g",  
                "AgentVersion": "1.0.7",  
                "TaskInstanceNumAvailable": 1,  
                "Cpu": 1  
            },  
            {  
                "ComputeNodeId": "node-c4z8f8xc",  
                "ComputeNodeState": "RUNNING",  
                "Mem": 2,  
                "ResourceCreatedTime": "2017-11-27T07:10:41Z",  
                "ComputeNodeInstanceId": "ins-fgqcau4q",  
                "AgentVersion": "1.0.7",  
            }  
        ]  
    }  
}
```

```
"TaskInstanceNumAvailable": 1,  
"Cpu": 1  
}  
],  
"CreateTime": "2017-11-27T07:10:02Z"  
}  
}
```

包含了计算环境整体，以及每个节点的详细信息。

销毁计算环境

请求示例

```
qcloudcli batch DeleteComputeEnv --Version 2017-03-12 --EnvId env-c96rwhnf
```

返回示例

```
{  
"Response": {  
"RequestId": "389f011a-7dbd-4993-82fe-334ac923ff88"  
}  
}
```

调用后计算环境会自动销毁集群内所有的云服务器。