

腾讯云批量计算

快速入门

产品文档



腾讯云

【版权声明】

©2013-2017 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

文档声明.....	2
快速入门.....	4
开始前的准备.....	4
如何开通批量计算？.....	5
命令行快速开始.....	6
命令行快速开始-计算环境.....	12
作业配置说明.....	19
控制台快速开始.....	26

快速入门

开始前的准备

1.创建腾讯云账号

如果您还没有腾讯云账号，请登录到[腾讯云官方网站](#)，点击右上角的“注册”创建腾讯云账号，详情可参考文档：[如何注册腾讯云](#)

2.开通批量计算

批量计算目前处于内测阶段，需要申请测试资格。请在[批量计算内测申请页](#)根据页面提示填写申请，请务必完整和准确的填写页面列举的选项。提交申请后我们的专业人员将与你联系，在之后的人工审核结束后，将会以短信的形式通知你开通结果。

3.获取 SecretId 和 SecretKey

开通了批量计算服务以后，用户若需要通过云API、SDK、CLI来操作批量计算的接口，需要使用 SecretId 和 SecretKey，请在 [API 密钥控制台](#) 里创建或查看 SecretId 和 SecretKey。

4.理解基本概念

为了使得用户能够对本产品进行更好地操作，用户有必要对本产品的一些[名词概念](#)进行了解，也可以通过[产品介绍页](#)获取产品信息和资料索引。

5.关于 COS

批量计算服务的标准输出和远程存储映射都与 COS 有关联，所以用户 务必对 [COS](#) 有一定的了解。

如何开通批量计算？

1. 访问申请页

批量计算（Batch）目前处于内测阶段，用户可以访问[批量计算Batch申请](#)来申请使用资格。

2. 填写申请信息

业务信息

行业类型* 基因测序 医疗分析 科学研究 视频渲染 金融大数据 广告和商业服务 其他

业务阶段* 未开发 开发中 测试中 已上线 大规模应用中

批量计算服务熟悉程度* 未使用过 自有机房试用 自有机房大规模使用 云上试用 云上大规模使用

常用的大数据计算服务器配置* 2核 4G 4核 8G 8核 16G 16核 32G 16核 32G 以上

批处理任务最多同时运行主机数量* 1-50台 51-200台 201-500台 501-1000台 1000台以上

业务简介*

精准医疗

提交申请

请务必完整和符合实际的填写相关信息，信息的完整和准确性将影响审核进度的快慢和通过与否。

3. 等待审核结果

申请提交后，将有专人与你联系并审核是否通过，审核结果将以短信的形式通知你。

命令行快速开始

一个最简单的例子

本例介绍如何使用命令行来提交一个简单的作业，例子是用 Python 实现的斐波那契数列求和，代码直接由 Batch 提供的任务命令配置部分输入，结果直接输出到任务配置的 stdout 输出地址上。

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）和对象存储（COS），用户需要先安装和配置命令行工具以及创建一个 COS Bucket。

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
```

```
DescribeAvailableCvmInstanceTypes |DescribeTask
```

```
DescribeJob |SubmitJob
```

```
DescribeJobs |TerminateTaskInstance
```

创建保存结果的COS Bucket

这个简单例子里，结果直接输出到系统标准输出里，Batch 可以采集系统标准输出 stdout 和 stderr，并在任务结束后将信息上传到你指定的 COS Bucket 里，你需要提前准备好保存信息的 Bucket 和子文件夹。

请参照 [命令行工具 - 前置准备](#) 里第三章节『准备 COS 目录』，创建对应 COS Bucket 和子文件夹。

作业配置简介

你可以从官方提供的例子来修改出一个在你的账号下可执行的 Batch 作业配置，在这之前先看一下作业的配置项各代表什么含义。

```
qcloudcli batch SubmitJob --Version 2017-03-12 --Job '{
  "JobName": "TestJob", // 作业名称
  "JobDescription": "for test ", // 作业描述
  "Priority": "1", // 作业优先级
  "Tasks": [ // 任务列表（本例仅一个任务）
  {
    "TaskName": "Task1", // 任务1名称
    "Application": { // 任务执行命令
      "DeliveryForm": "LOCAL", // 执行本地命令
      "Command": "python -c \"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(fib(20))\" " //
命令具体内容（斐波拉契求和）
    },
    "ComputeEnv": { // 计算环境配置
      "EnvType": "MANAGED", // 计算环境类型，托管型和非托管型
      "EnvData": { // 具体配置（当前托管型，可参照CVM 创建实例说明）
        "InstanceType": "S1.SMALL1", // CVM 实例类型
        "ImageId": "img-m4q71qnf", // CVM 镜像 ID（需替换）
      }
    },
    "RedirectInfo": { // 标准输出重定向配置
      "StdoutRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/", //
标准输出（需替换）
      "StderrRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/" //
标准错误（需替换）
    }
  }
]
}'
--Placement'{
  "Zone": "ap-guangzhou-2" // 可用区（可能需替换）
}'
```

Batch 的 SubmitJob 命令包含 3 个参数

- Version : 版本号, 目前固定填写 2017-03-12
- Job : 作业配置, JSON 格式, 详细字段意义见示例
- Placement : 执行作业的可用区

* 1. Job 里标识需替换的字段, 需要替换成用户自己的信息后才可以执行, 比如自定义镜像 Id, VPC 相关信息, COS Bucket 地址和对应 SecretId、SecretKey。

* 2. 上面的例子加入了注释文字, 因此无法直接在命令行工具运行, 请复制下方的示例然后填写完『待填写』字段后再执行。命令较长, 请使用后方复制按钮防止复制不完整。

* 3. 详细 Job 配置说明请参考

[作业配置说明](#)

。

```
qcloudcli batch SubmitJob --Version 2017-03-12 --Job '{"JobName": "TestJob", "JobDescription": "for test", "Priority": "1", "Tasks": [{"TaskName": "Task1", "TaskInstanceNum": 1, "Application": {"DeliveryForm": "LOCAL", "Command": "python -c \"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(fib(20))\" \"}, "ComputeEnv": {"EnvType": "MANAGED", "EnvData": {"InstanceType": "S1.SMALL1", "ImageId": "待替换" }}, "RedirectInfo": {"StdoutRedirectPath": "待替换", "StderrRedirectPath": "待替换"}, "MaxRetryCount": 1 } ]}' --Placement '{"Zone": "ap-guangzhou-2"}
```

修改配置

1. 填写 ImageId

```
"ImageId": "待替换"
```

内测需要使用基于 Cloud-init 服务和配置过的镜像，官方提供 CentOS 6.5 的可直接使用镜像，镜像 ID 是 img-m4q71qnf，Windows Server 2012 的官方镜像 ID 是 img-iy8pcwen。

2. 配置 StdoutRedirectPath 和 StderrRedirectPath

```
"StdoutRedirectPath": "待替换", "StderrRedirectPath": "待替换"
```

将你在前置准备里创建 COS Bucket 的访问地址填写到 StdoutRedirectPath 和 StderrRedirectPath 中。

3. 修改可用区（可选）

```
--Placement '{"Zone": "ap-guangzhou-2"}'
```

例子中指定在广州二区申请资源，你可以根据你在命令行工具配置的默认地域，选择相应的可用区申请资源，地域和可用区的详细信息请查看[地域和可用区>>](#)。

4. Windows下命令行输入JSON格式修改（可选）

Windows 下的命令行工具输入 JSON 格式数据会区别于 Linux，比如 " 号需要替换成 \，详细请见[快速使用腾讯云命令行工具](#)的『JSON 格式作为入参』章节。

查看结果

```
{
  "Response": {
    "RequestId": "5d928636-bba2-4ab3-bef3-cf17d7c73c51",
    "JobId": "job-1rqdgnqn"
  }
}
```

返回 JobId 则表示执行成功。

```
$ qcloudcli batch DescribeJob --Version 2017-03-12 --JobId job-1z4yl2bp
{
  "Response": {
    "JobState": "STARTING",
    "Zone": "ap-guangzhou-2",
    "JobName": "test job",
    "Priority": 1,
    "RequestId": "b116f9b5-410c-4a69-bbe8-b695a2d6a869",
    "TaskSet": [
      {
        "TaskName": "hello2",
        "TaskState": "STARTING"
      }
    ],
    "JobId": "job-1z4yl2bp",
    "DependenceSet": []
  }
}
```

可以通过 DescribeJob 来查看刚才提交的任务信息。

```
$ qcloudcli batch DescribeJobs --Version 2017-03-12
```

还可以通过 DescribeJobs 查看当前地域作业列表。

下一步可以干什么？

这个是一个最简单的例子，它是一个单任务的作业，也没有使用到远程存储映射能力，仅仅是向用户展示最基

本的能力，你可以根据 API 说明文档继续测试 Batch 更高阶的能力。

- 更简单的操作方法：Batch 的能力强大，配置项较多，通过脚本来调用会更加简便快捷，从 [前置准备](#) 和 [1_简单开始](#) 开始尝试这种方式。
- 执行远程代码包：Batch 提供自定义镜像 + 远程代码包 + 命令行的方式，在技术上全方位的覆盖您的业务需要，详细查看 [2_执行远程代码包](#)。
- 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，详细查看 [3_远程存储映射](#)。

命令行快速开始-计算环境

一个使用计算环境的例子

本例介绍如何使用命令行来创建计算环境，然后向指定计算环境提交一个作业，最后销毁计算环境。

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）和对象存储（COS），用户需要先安装和配置命令行工具以及创建一个 COS Bucket。

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
```

创建保存结果的COS Bucket

这个简单例子里，结果直接输出到系统标准输出里，Batch 可以采集系统标准输出 stdout 和 stderr，并在任务结束后将信息上传到你指定的 COS Bucket 里，你需要提前准备好保存信息的 Bucket 和子文件夹。

请参照 [命令行工具 - 前置准备](#) 里第三章节『准备 COS 目录』，创建对应 COS Bucket 和 子文件夹。

创建计算环境

你可以从官方提供的例子来修改出一个在你的账号下可执行的 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。你也可以参考 [创建计算环境](#) 等计算环境相关接口。

```
qcloudcli batch CreateComputeEnv --Version 2017-03-12 --ComputeEnv '{
  "EnvName": "test compute env", // 计算环境名称
  "EnvDescription": "test compute env", // 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）
    "InternetAccessible": {
      "PublicIpAssigned": "TRUE",
      "InternetMaxBandwidthOut": 50
    },
    "LoginSettings": {
      "Password": "B1[habcd"
    },
    "InstanceType": "S1.SMALL1", // CVM 实例类型
    "ImageId": "img-xxxxyyyy" // CVM 镜像 ID（需替换）
  },
  "DesiredComputeNodeCount": 2 // 计算节点期望个数
}'
--Placement'{
  "Zone": "ap-guangzhou-2" // 可用区（可能需替换）
}'
```

请求示例

```
qcloudcli batch CreateComputeEnv --Version 2017-03-12 --ComputeEnv '{"EnvName": "test compute env", "EnvDescription": "test compute env", "EnvType": "MANAGED", "EnvData": {"InstanceType":
```

```
"S1.SMALL2", "ImageId": "待替换", "LoginSettings": {"Password": "待替换"}, "InternetAccessible":  
{ "PublicIpAssigned": "TRUE", "InternetMaxBandwidthOut": 50}, "SystemDisk": {"DiskType":  
"CLOUD_BASIC", "DiskSize": 50 }}, "DesiredComputeNodeCount": 2 }' --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"  
  }  
}
```

查看计算环境列表

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"
}
}
```

查看指定计算环境

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-fgqc4u4q",
    "AgentVersion": "1.0.7",
    "TaskInstanceNumAvailable": 1,
    "Cpu": 1
  }
]
```

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

向指定计算环境提交任务

请求示例

```
qcloudcli batch SubmitJob --Version 2017-03-12 --Job '{"JobName": "test job", "JobDescription":  
"xxx", "Priority": "1", "Tasks": [{"TaskName": "hello2", "TaskInstanceNum": 1, "Application":  
{"DeliveryForm": "LOCAL", "Command": "python -c \"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2);  
print(fib(20))\" "}, "EnvId": "待替换", "RedirectInfo": {"StdoutRedirectPath":  
"cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/", "StderrRedirectPath":  
"cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/"} } ]}' --Placement '{"Zone": "ap-  
guangzhou-2"}'
```

返回示例

```
{  
  "Response": {  
    "RequestId": "cf821ae0-71d6-42b1-b878-6ecdeec15796",  
    "JobId": "job-lhi5agkh"  
  }  
}
```

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"  
  }  
}
```

作业配置说明

1. 简要说明

批量计算 Batch 的作业配置以 JSON 格式提供，下面给出这个配置的简要说明，下面的作业包含 2 个任务

```
{
  "JobName": "TestJob", // 作业名称
  "JobDescription": "for test ", // 作业描述
  "Priority": "1", // 作业优先级
  "Tasks": [ // 任务列表（本例包含两个任务）
    {
      // 任务1（最简化的任务配置，去除所有非必选项）
      "TaskName": "Task1", // 任务1名称
      "Application": { // 任务执行命令
        "DeliveryForm": "LOCAL", // 应用程序的交付方式
        "Command": "echo hello" // 命令具体内容（输出 hello）
      },
      "ComputeEnv": { // 计算环境配置
        "EnvType": "MANAGED", // 计算环境类型，托管型和非托管型
        "EnvData": { // 具体配置（当前托管型，可参照CVM 创建实例说明）
          "InstanceType": "S1.SMALL1", // CVM 实例类型
          "ImageId": "img-m4q71qnf", // CVM 镜像 ID
        }
      },
      "RedirectInfo": { // 标准输出重定向配置
        "StdoutRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/", //
        // 标准输出（需替换）
        "StderrRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/" //
        // 标准错误（需替换）
      },
      "Authentications": [ // 鉴权相关信息（选填，访问非本人COS场景使用）
        {
```

```
"Scene": "COS", // 场景 (当前是 COS)
"SecretId": "****", // SecretId (需替换)
"SecretKey": "****" // SecretKey (需替换)
}
]
},
{
// 任务2
"TaskName": "Task2", // 任务2名称
"TaskInstanceNum": 1, // 任务2并发实例数目
"Application": { // 任务执行命令
"DeliveryForm": "LOCAL", // 执行本地命令
"Command": "python -c \"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(fib(20))\" " //
命令具体内容 (斐波拉契求和)
},
"ComputeEnv": { // 计算环境配置
"EnvType": "MANAGED", // 计算环境类型, 托管型和非托管型
"EnvData": { // 具体配置 (当前托管型, 可参照CVM 创建实例说明)
"InstanceType": "S1.SMALL1", // CVM 实例类型
"ImageId": "img-m4q71qnf", // CVM 镜像 ID (可替换)
"VirtualPrivateCloud": { //□ CVM 网络配置 (选填)
"VpcId": "vpc-cg18la4l", // VpcId (需替换)
"SubnetId": "subnet-8axej2jc" // SubnetId (需替换)
},
"SystemDisk": { //□ CVM 系统盘配置
"DiskType": "CLOUD_BASIC",
"DiskSize": 50
},
},
"DataDisks": [ //□ CVM 数据盘配置
{
"DiskType": "CLOUD_BASIC",
"DiskSize": 50
}
]
}
```

```
}  
},  
"RedirectInfo": { // 标准输出重定向配置  
  "StdoutRedirectPath": "cos://dondonbatchv5- 1251783334.cosgz.myqcloud.com/logs/", //  
  标准输出 (需替换)  
  "StderrRedirectPath": "cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/" //  
  标准错误 (需替换)  
},  
"MaxRetryCount": 1, // 最大重试数目  
"Authentications": [ // 鉴权相关信息 (选填, 访问非本人COS场景使用)  
{  
  "Scene": "COS", // 场景 (当前是 COS)  
  "SecretId": "****", // SecretId (需替换)  
  "SecretKey": "****" // SecretKey (需替换)  
}  
]  
},  
"Dependences": [  
{  
  "StartTask": "Task1",  
  "EndTask": "Task2"  
}  
]  
}
```

2. 详细说明

I. 作业 (Job)

作业是 Batch 提交的单元, 除了本身信息, 还包含了一个或者多个任务 (Task) 的信息以及 Task 之间的依赖关系

名称	类型	是否必选	描述
JobName	String	否	作业名称
JobDescription	String	否	作业描述
Priority	Integer	是	作业优先级，任务（Task）和任务实例（TaskInstance）会继承作业优先级
Tasks.N	array of Task objects	是	任务信息
Dependences.N	array of Dependence objects	否	依赖信息

II. 任务（Task）

一个作业可以包含多个任务，任务主要描述了批处理数据计算中，实际计算过程依赖的环境（机型、系统、镜像）、执行的代码包和命令行、存储、网络等相关信息。

名称	类型	是否必选	描述	示例
TaskName	String	是	任务名称，在一个作业内部唯一	Task1
TaskInstanceNum	Integer	是	任务实例运行个数	1
Application	Application object	是	应用程序信息	
ComputeEnv	ComputeEnv object	是	运行环境信息	
RedirectInfo	RedirectInfo object	是	重定向路径	
InputMappings	array of InputMapping object	否	输入映射	
OutputMappings	array of OutputMapping object	否	输出映射	
Authentications	array of Authentication object	否	授权信息	
MaxRetryCount	Integer	否	任务失败后的最大重试次数	3
Timeout	Integer	否	任务启动后的超时时	3600

名称	类型	是否必选	描述	示例
			间，单位秒	

Application

名称	类型	是否必选	描述	示例
Command	String	是	任务执行命令	
DeliveryForm	String	是	应用程序的交付方式	LOCAL 本地，PACKAGE 远程代码包
PackagePath	String	否	远程代码包路径，必须 .tgz 格式	http://batchdemo-1251783334.cosgz.myqcloud.com/codepkg/codepkg.tgz (仅 PACKAGE 方式)

ComputeEnv

名称	类型	是否必选	描述	示例
EnvType	String	是	计算环境管理类型，包括托管和非托管两种	LOCAL 本地，PACKAGE 远程代码包)
EnvData	EnvData object	是	计算环境具体参数	

EnvData

名称	类型	是否必选	描述	示例
InstanceType	String	是	CVM实例类型，托管类型必填	img-m4q71qnf
ImageId	String	是	CVM镜像ID，托管类型必填	S1.SMALL1
others	others	否	参考 CVM API文档 创建实例 提供的参数	支持 SystemDisk、DataDisks、VirtualPrivateCloud 等)

RedirectInfo

名称	类型	是否必选	描述	示例
StdoutRedirectPath	String	否	标准输出重定向路径	cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/
StderrRedirectPath	String	否	标准错误重定向路径	cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/logs/

InputMapping

名称	类型	是否必选	描述	示例
SourcePath	String	是	源端路径	cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/input/
DestinationPath	String	是	目的端路径	/data/input/

OutputMapping

名称	类型	是否必选	描述	示例
SourcePath	String	是	源端路径	/data/output/
DestinationPath	String	是	目的端路径	cos://dondonbatchv5-1251783334.cosgz.myqcloud.com/output/

Authentication

如果填写的 COS 路径（存储映射、日志重定向）是本人 COS 地址，无需填写。需要访问其他人的 COS 时，需要填写对应的访问密钥。

名称	类型	是否必选	描述
Scene	String	是	授权场景，例如COS

名称	类型	是否必选	描述
SecretId	String	是	SecretId
SecretKey	String	是	SecretKey

III. 任务依赖 (Dependence)

描述任务之间的先后关系，假设作业包含 2 个任务， StartTask 为 Task1， EndTask 为 Task2，则会在执行完 Task1 之后才会启动 Task2，Task2 执行完则作业执行完毕。

名称	类型	是否必选	描述	示例
StartTask	String	是	依赖关系的起点任务名称	Task1
EndTask	String	是	依赖关系的起点任务名称	Task2

控制台快速开始

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，具体操作步骤如下。

准备

准备好 [对象存储](#) 存储桶。如果您尚未创建存储桶，请参照 [创建存储桶](#) 完成创建。

登录[控制台](#)

如果您尚未开通批量计算服务，请参照批量计算控制台主页相关提示开通。

创建任务模板

1. 单击左侧导航栏“任务模板”选项，选择目标地域，比如“广州”。点击【新建】按钮。
2. 配置基本信息。
 - 名称：如 hello
 - 描述：如 hello demo
 - 资源配置：默认值
 - 资源数量：如 1 台
 - 超时时间：默认值
 - 重试次数：默认值
 - 镜像：img-m4q71qnf

基本信息

名称

hello

描述

echo 'hello, world'

资源配置

S1.SMALL1 (1核1GB) ▾

[云主机详细配置](#)

系统盘 (50GB)、带宽 (无外网带宽)、密码 (系统自动生成)

资源数量

−

1

+

台

超时时间 ⓘ

259200

秒

重试次数 ⓘ

0 ▾

镜像

img-i91njcmx

下一步

3. 配置程序信息。

- 执行方式：Local
- Stdout日志：格式参考 [COS、CFS路径填写](#)
- Stderr日志：同 Stdout 日志
- 命令行：echo 'hello, world'

程序配置

执行方式

Stdout日志 校验通过 校验

Stderr日志 校验

命令行

4. 配置存储映射，完成后单击【下一步】按钮。

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
添加	

5. 预览任务 JSON 文件，确认无误后，单击【保存】按钮。

任务模板JSON文件预览

```

1 {
2   "showDialog": false,
3   "cvmIptVal": "",
4   "showPwd": false,
5   "TaskTemplateInfo": {
6     "Timeout": 259200,
7     "MaxRetryCount": 0,
8     "TaskInstanceNum": 1,
9     "Application": {
10      "Command": "echo 'hello, world'",
11      "DeliveryForm": "LOCAL"
12    },
13    "ComputeEnv": {
14      "EnvType": "MANAGED",
15      "EnvData": {
16        "InstanceType": "S1.SMALL1",
17        "ImageId": "img-i91njcmx",
18        "SystemDisk": {
19          "DiskType": "CLOUD_BASIC",
20          "DiskSize": 50
21        },
22        "DataDisks": [
23          {
24            "DiskType": "CLOUD_BASIC",
25            "DiskSize": 0
26          }
27        ]
28      }
29    }
30  }
31 }

```

上一步
保存

6. 查看任务模板。

ID/名称	备注	创建时间	操作
task-tmpl-nuydksjg hello	echo 'hello, world'	2018-2-27 10:37:47	删除

提交作业

1. 单击左侧导航栏“作业”选项，选择目标地域，比如“广州”。点【新建】按钮。
2. 配置作业基本信息。

- 作业名称：如 hello
- 优先级：默认值
- 描述：如 hello job

作业名称

优先级

范围 0-100，数值越大，优先级越高

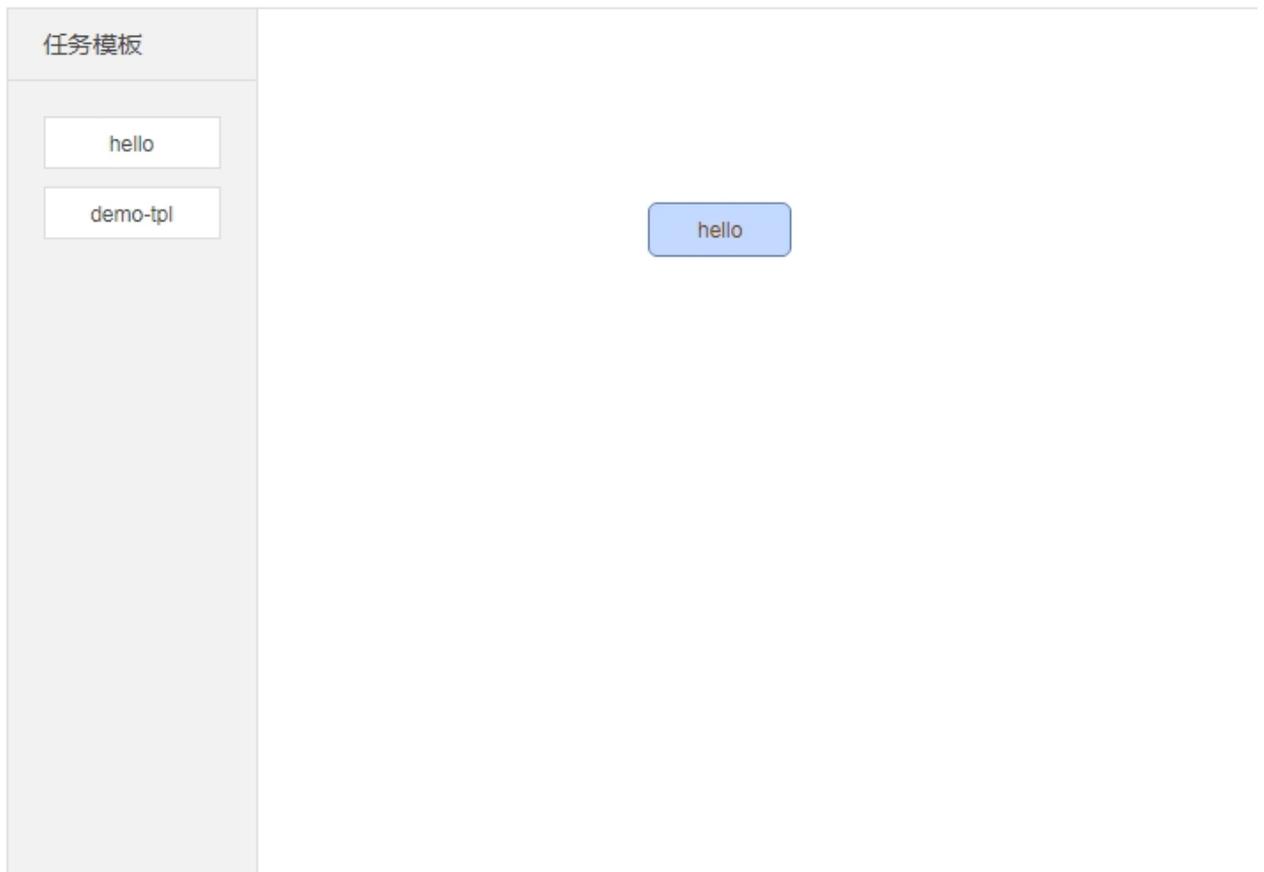
描述

3. 选中“任务流”左侧“hello”任务，移动鼠标将任务放置到右侧画布中。

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



4. 打开“任务流”右侧“任务详情”，确认配置无误后，点击“完成”按钮。

任务信息

hello

基本信息

名称

资源配置 ▼

系统盘 (50GB)、数据盘 (0GB) ⓘ

并发实例数

命令行

echo 'hello, world'

ID task-tmpl-nuydksjg

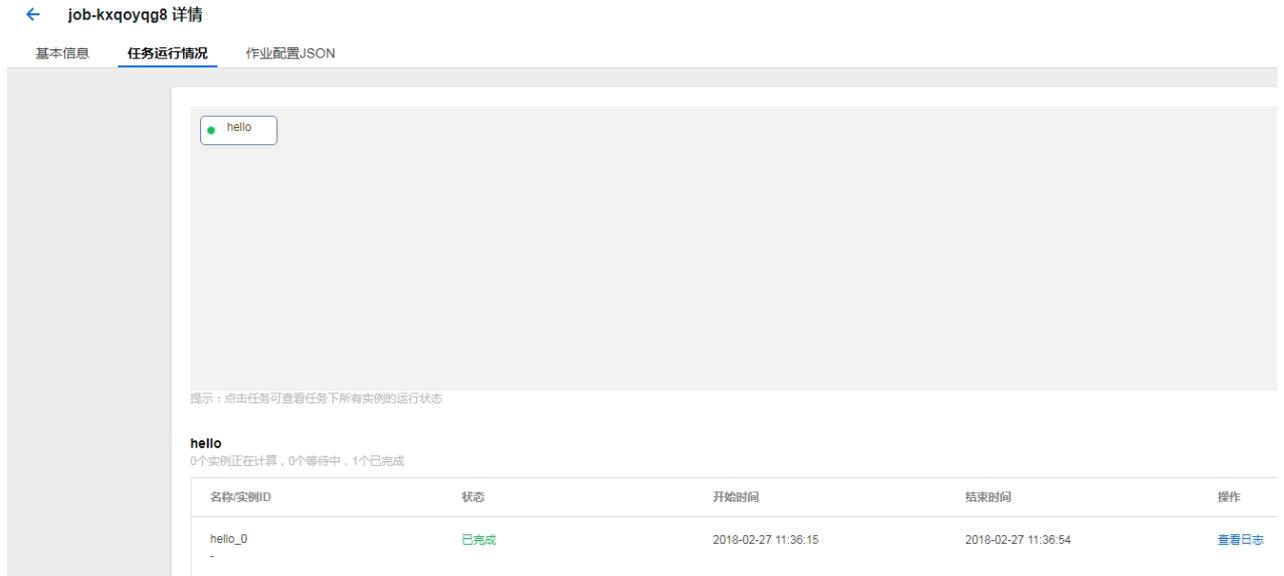
创建时间 2018-02-27 10:37:47

操作 ...

5. 查询结果。您可以在作业列表页查看作业的运行状态。

ID/名称	状态	已完成/总任务数	开始时间	结束时间	操作
job-lxqoyag8 hello	成功	1/1	2018-02-27 11:35:05	2018-02-27 11:36:54	删除

- 点击作业ID，在“任务运行情况”下可以看到各个任务实例的运行状态
- 点击“查询日志”按钮，可以查看任务实例的标准输出和标准错误。



下一步可以干什么？

这个是一个最简单的例子，它是一个单任务的作业，也没有使用到远程存储映射能力，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。

- 丰富的云主机配置：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。
- 执行远程代码包：Batch 提供 自定义镜像 + 远程代码包 + 命令行 的方式，在技术上全方位的覆盖您的业务需要。
- 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。