

批量计算 快速入门 产品文档



腾讯云

【 版权声明 】

©2013–2021 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

快速入门

开始前的准备

命令行快速开始

命令行快速开始-计算环境

作业配置说明

控制台快速开始

快速入门

开始前的准备

最近更新时间：2020-09-11 16:13:40

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](#)，单击右上角的【注册】创建腾讯云账号，详情可参考文档：[如何注册腾讯云](#)。

2. 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，则要使用到 SecretId 和 SecretKey，请在 [API 密钥控制台](#) 里创建或查看 SecretId 和 SecretKey。

3. 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。

4. 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，所以需要用户对 [COS](#) 有一定的了解。

命令行快速开始

最近更新时间：2020-10-23 17:16:44

操作场景

本文介绍如何使用命令行 TCCLI 来提交一个简单的作业，文中示例为使用 Python 实现对斐波那契数列求和。Python 代码由任务中 Application 参数的 Command 字段指定，返回结果直接输出到任务配置的 stdout 输出地址。

前提条件

请参考 [开始前的准备](#) 中的步骤做好准备。

操作步骤

安装和配置 TCCLI

1. 请参考 [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。

创建保存结果的 COS Bucket

在本文示例中，返回结果将直接输出到系统标准输出中，而 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已指定的 COS Bucket 中，您需提前创建 Bucket 及用于保存信息的子文件夹。

请参考 [准备 COS 目录](#) 创建对应 COS Bucket 和子文件夹。

作业配置简介

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。请参考以下内容了解计算环境各项配置的含义：

```
tccli 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）
}'
```

SubmitJob 命令

Batch 的 SubmitJob 命令示例：

```
tccli 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"}'
```

SubmitJob 命令包含以下3个参数：

参数	描述
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

- 在示例中已标识需替换的字段，您需要替换为个人信息后才可以执行。例如自定义镜像 ID，VPC 相关信息，COS Bucket 地址和对应 SecretId、SecretKey，可参考 [修改配置](#)。
- 请使用示例后方的复制按钮，防止复制内容不完整。请将待替换信息全部替换后再执行命令。
- 详细 Job 配置说明请参考 [作业配置说明](#)。

修改配置

填写 ImageId

```
"ImageId": "待替换"
```

您需要使用基于 Cloud-init 服务和配置过的镜像，官方提供可直接使用镜像如下：

- CentOS 6.5 镜像：ID 为 `img-m4q71qnf`。
- Windows Server 2012 的官方镜像：ID 为 `img-er9shcln`。

配置 StdoutRedirectPath 和 StderrRedirectPath

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

请将在 [创建保存结果的 COS Bucket](#) 获取的访问域名填写到 StdoutRedirectPath 和 StderrRedirectPath 中。

修改可用区（可选）

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

示例中指定在广州二区申请资源，您可以根据 TCCLI 中配置的默认地域，选择相应的可用区并申请资源。地域和可用区的详细信息请查看 [地域和可用区](#)。

查看结果

返回结果如下，则表示执行成功。

```
{
  "RequestId": "db84b7f8-ff8e-4c11-81c1-9a3b02652a46",
  "JobId": "job-cr8qyyh9"
}
```

- 执行以下命令，通过 DescribeJob 查看刚才提交的任务信息。

```
$ tccli batch DescribeJob --version 2017-03-12 --JobId job-cr8qyyh9
{
  "EndTime": "2019-10-08T07:28:07Z",
  "JobState": "SUCCEED",
  "TaskInstanceMetrics": {
    ...
  },
  "Zone": "ap-guangzhou-2",
  "TaskMetrics": {
    ...
  },
  "JobName": "TestJob",
  "Priority": 1,
  "RequestId": "0e5c5ea5-ef25-4f90-b355-cfaa8057d473",
  "TaskSet": [
    {
      ...
    }
  ]
}
```

```
],  
  "StateReason": null,  
  "JobId": "job-cr8qyyh9",  
  "DependenceSet": [],  
  "CreateTime": "2019-10-08T07:27:17Z"  
}
```

- 执行以下命令，通过 `DescribeJobs` 查看当前地域作业列表。

```
$ tccli batch DescribeJobs --version 2017-03-12
```

更多功能

文中的示例为单任务的作业，仅展示了最基本的功能，未使用远程存储映射能力，您可以通过以下文档或参考 [API 文档](#) 深入了解并使用 Batch：

- **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，请参考 [前置准备](#) 和 [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，在技术上全方位覆盖您的业务需要，详情请参考 [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，详情请参考 [远程存储映射](#)。

命令行快速开始-计算环境

最近更新时间：2020-10-23 16:37:24

操作场景

本文介绍如何使用命令行 TCCLI 来创建计算环境、向计算环境提交作业、销毁计算环境。

前提条件

请参考 [开始前的准备](#) 中的步骤做好准备。

操作步骤

安装和配置 TCCLI

1. 请参考 [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。

创建保存结果的 COS Bucket

在本文示例中，返回结果将直接输出到系统标准输出中，而 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已指定的 COS Bucket 中，您需提前创建 Bucket 及用于保存信息的子文件夹。

请参考 [准备 COS 目录](#) 创建对应 COS Bucket 和子文件夹。

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。请参考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。

```
tccli 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": "*"*"*"*" // 登录密码（需替换）
    },
    "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

```
tccli 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

返回值如下所示，其中 EnvId 为 Batch 计算环境的唯一标识。

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

查看计算环境列表

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（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

请求示例

执行以下命令，查看指定计算环境。

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

返回示例（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"
}
```

向指定计算环境提交任务

请求示例

请结合您的实际情况，替换命令中的相关信息并执行，向指定计算环境提交任务。

```
tccli 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": "待替换", "StderrRedirectPath": "待替换"}}]}' --Placement '{"Zone": "ap-guangzhou-2"}'
```

返回示例

```
{  
  "RequestId": "d6903404-5765-474b-b516-39137456fa5a",  
  "JobId": "job-qjq3mqp7"  
}
```

销毁计算环境

请求示例

执行以下命令，销毁计算环境。

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

返回示例

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

作业配置说明

最近更新时间：2020-06-03 08:28:43

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

名称	类型	是否必选	描述	示例
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间，单位秒	3600

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
----	----	------	----	----

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	S1.SMALL1
ImageId	String	是	CVM镜像 ID，托管类型必填	img-m4q71qnf
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
----	----	------	----	----

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
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	依赖关系的终点任务名称	Task2

控制台快速开始

最近更新时间：2020-06-03 20:39:08

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. 选择左侧导航栏中的【任务模板】选项，并在页面上方选择目标地域，例如“广州”。

2. 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：

基本信息

名称	<input type="text" value="hello"/>
描述	<input type="text" value="hello demo"/>
计算环境类型 ^①	<input type="radio"/> 已有计算环境 <input checked="" type="radio"/> 自动计算环境
资源配置	<input type="text" value="S2.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)
镜像	<input type="text" value="公有镜像"/> <input type="text" value="CentOS 7.6 64位"/> 必须选择安装和配置过Cloud-init的镜像 ^①
资源数量	<input type="text" value="-"/> <input type="text" value="1"/> <input type="text" value="+"/> 台
超时时间 ^①	<input type="text" value="259200"/> 秒
重试次数 ^①	<input type="text" value="0"/>

下一步

主要参数信息如下：

- **名称**：自定义名称，例如 hello。
- **描述**：自定义描述，例如 hello demo。
- **计算环境类型**：
 - **已有计算环境**：可选择已有计算环境。
 - **自动计算环境**：即无须预先创建固定的计算环境。作业提交后，自动创建 CVM 实例并运行任务，任务完成后自动销毁实例。
- **镜像**：请按需选择。
- **资源数量**：例如1台。
- **超时时间、重试次数**：保持默认值。

3. 单击【下一步】，配置程序信息。如下图所示：

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

- **执行方式**：选择“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'。

4. 单击【下一步】，配置存储映射。如下图所示：

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. 单击【下一步】，预览任务 JSON 文件，确认无误后单击【保存】。如下图所示：

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. 成功创建后，即可在“任务模板”页面进行查看。如下图所示：

任务模板

广州

批量计算帮助文档

ID/名称	备注	创建时间	操作
task- img-i91njcmx test1		2020-4-13 15:9:11	删除

提交作业

1. 选择左侧导航栏中的【作业】，并在页面上方选择目标地域，例如“广州”。

2. 单击【新建】，进入“新建作业”页面，参考以下信息配置作业信息。如下图所示：

作业名称

优先级

范围 0-100，数值越大，优先级越高

描述

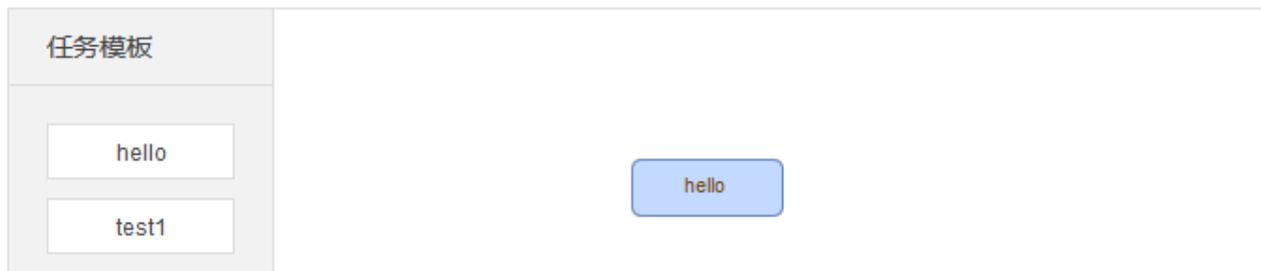
- **作业名称**：自定义名称，例如 hello。
- **优先级**：保持默认值。
- **描述**：自定义描述，例如 hello demo。

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。如下图所示：

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. 在“任务流”右侧“任务详情”，确认配置无误后，单击【完成】。

5. 您可以在“作业”列表页查看作业的运行状态。如下图所示：

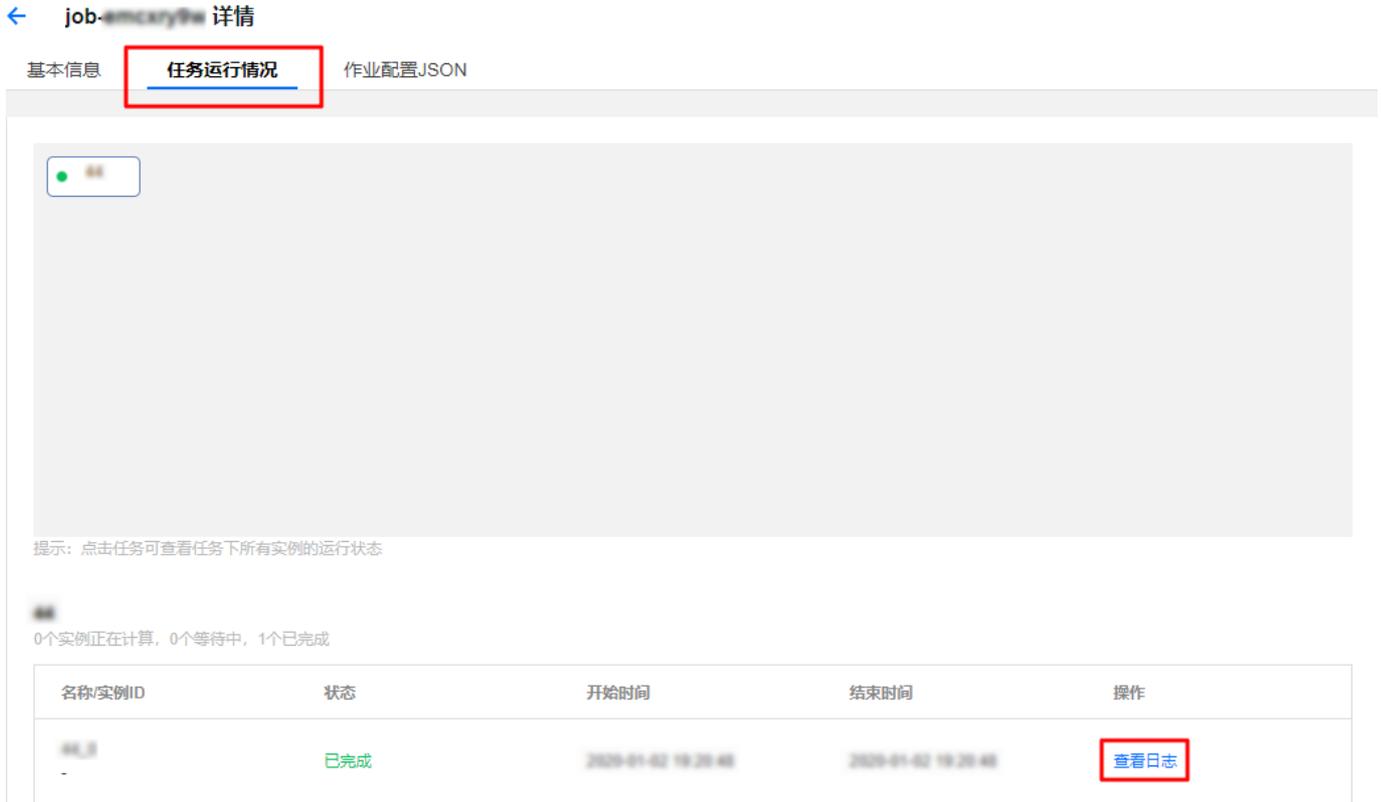
作业 广州 批量计算帮助文档

新建

ID/名称	状态	已完成/总任务数	开始时间	结束时间	操作
job- id77h49 hello	启动中	0/1	2020-06-03 20:30:57	-	终止

- 单击作业 ID，在【任务运行情况】页签下，可查看到个任务实例的运行状态。

- 单击【查询日志】，可查看任务实例的标准输出和标准错误。如下图所示：



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。
- 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。