

Alibaba Cloud

函数计算
场景案例

文档版本：20220705

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.使用函数模板	05
1.1. 模板列表	05
1.2. 使用模板创建函数	10
2.基于函数计算和TensorFlow的Serverless AI推理	13

1.使用函数模板

1.1. 模板列表

函数计算为您提供了多种示例场景的模板代码，您可以在示例代码上进行更改，也可以直接使用模板代码。本文介绍函数模板的使用说明与参数配置。

背景信息

函数计算提供了多种函数模板，您可以在[函数计算控制台](#)创建函数页面的使用模板创建页签内，查看模板详细信息。部分模板是需要配置对应触发器后才能使用，例如备份OSS增量数据等。您可以参考以下模板列表，查看相应信息：

- [alimebot-nodejs](#)：云小蜜机器人
- [image-crawler](#)：图片爬虫
- [image-recognition](#)：图像识别
- [flask-web](#)：基于Flask工程部署到函数计算
- [load-large-code-package](#)：加载大代码包
- [aliyun-config-python3](#)：审计规则配置
- [api-gateway-nodejs12](#)：API网关返回串大全
- [copy-oss-object-python3](#)：备份OSS增量数据
- [data-lake-analytics](#)：数据湖实践
- [get-object-meta](#)：获取OSS对象属性
- [get-oss-md5-python3](#)：获取OSS文件MD5
- [logstore-replication](#)：日志备份
- [marketplace-weather-demo](#)：使用云市场服务
- [oss-download-zip-upload](#)：OSS图片下载压缩
- [oss-get-object-info](#)：获取对象信息
- [oss-shipper-csv](#)：构建CSV投递OSS
- [read-write-tablestore-nodejs12](#)：读写表格存储
- [read-write-tablestore-python3](#)：读写表格存储
- [return-apigateway-string](#)：API网关返回串

[alimebot-nodejs](#)：云小蜜机器人

该模板供Dialog Studio的开发者使用，提供了基本的函数结构，您需要按此结构进行代码开发。

- 输入参数：
 - event：由上游节点传入的参数信息。
 - context：上下文信息，例如对话。
- 输出参数：result。通过Callback的异步方式读取返回结果。

[image-crawler](#)：图片爬虫

该模板可以根据您提供的URL地址，抓取该地址中所有JPG格式的图片文件，并转存到OSS中。

② 说明 使用该函数前，您需要完成以下操作：

- 开通[日志服务SLS](#)和[对象存储OSS](#)。
- 在ServiceRole中添加AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：
 - URL：任意包含JPG格式图片的URL。
 - EndPoint：Bucket的EndPoint。
 - BucketName：Bucket名称。
- 输出参数：将URL上所有JPG格式的文件上传到OSS指定的Bucket中。
 - ok：转存成功。
 - error：转存出错。

image-recognition：图像识别

当您通过OSS上传图片时，您可以触发函数计算服务进行图像识别，并且按照识别的结果进行分类存储分析。

② 说明 使用该函数前，您需要完成以下操作：

- 开通[日志服务SLS](#)和[对象存储OSS](#)。
- 在ServiceRole中添加AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：
 - EndPoint：Bucket的EndPoint。
 - BucketName：Bucket的名称。
- 输出参数：将解析识别的结果，归类存储到OSS不同的Bucket中。
 - ok：成功。
 - error：出错。

flask-web：基于Flask工程部署到函数计算

根据该模板提供的示例代码，您可以通过URL访问函数，实现Flask Web工程的Serverless。

- 输入参数：通过HTTP触发。
- 输出参数：Web HTML页面。

load-large-code-package：加载大代码包

此模版使用initializer接口解决函数计算上传代码包时的大小受限问题。当您的代码包过大或超过函数计算系统上限时，可以预先将一部分代码包上传到OSS中，并在initializer中加载代码包。函数计算保证在处理函数逻辑执行之前，Initializer函数有且仅成功执行一次，从而可以保证在处理请求到达之前，代码已全部加载完成，并不会对处理函数逻辑的功能和性能产生影响。

 **说明** 使用该函数前，您需要完成以下操作：

- 为Initializer函数加载代码配置以下环境变量：
 - Bucket：代码包所在的OSS Bucket。
 - CodeObject：代码包Object名称。
 - Endpoint：OSS代码包所在地域的Endpoint。
 - CodePath：从OSS下载解压后存放的目录。
- 在ServiceRole中添加AliyunOSSReadOnlyAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：自定义。
 - initializer：初始化函数。
 - handler：处理函数。

- 输出参数：自定义。

aliyun-config-python3：审计规则配置

该模板为配置审计产品的自定义规则模板，方便您快速开发企业定制化的合规规则，实现持续的基础设施的合规监管。

api-gateway-nodejs12：API网关返回串大全

该模板是一个API网关的后端实现，您可以通过该函数获取不同的响应内容，包括HTML页面、JSON和图片。

- 输入参数：

```
{
  "queryParameters": {
    "response": "html|json|image"
  }
}
```

- 输出参数：HTML页面、JSON或图片。

```
html, or json, or image
```

copy-oss-object-python3：备份OSS增量数据

通过该模板的示例代码，您可以把指定某个Bucket下的文件备份到其他资源。

 **说明** 在ServiceRole中添加AliyunOSSReadOnlyAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：通过OSS事件触发。
- 输出参数：异步执行。

data-lake-analytics：数据湖实践

该模板通过函数计算调用Serverless化的Data Lake Analytics，进行OSS数据查询分析并将结果回流到RDS for MySQL。函数计算与Data Lake Analytics的结合，能够轻松实现云上轻量级的数据调度与处理。更多信息，请参见[基于Data Lake Analytics的数据湖实践](#)。

- 输入参数：自定义。

- 输出参数：自定义。

get-object-meta：获取OSS对象属性

您可以上传一个指定前缀或后缀的Object到Bucket，函数计算通过event参数获取地域、Bucket名称和Object名称，然后把获取到的Object的Meta信息写入一个文件，并将该文件上传到指定的Bucket。

 **说明** 使用该函数前，您需要完成以下操作：

- 设置OSS触发器，触发事件为oss: ObjectCreated: PostObject。更多信息，请参见[PostObject](#)。
- 在ServiceRole中添加GetObject、PutObject、AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：通过OSS的Post Object接口上传Object然后触发函数执行，函数可以从传入的event参数中获取您上传Object的地域、Bucket名称和Object名称等信息。
- 输出参数：产生一个文件用于记录被上传Object的Meta信息，并上传该文件到OSS指定的Bucket中。

get-oss-md5-python3：获取OSS文件MD5

通过该模板的示例代码，您可以计算出OSS文件的MD5值。通过流式方式读取文件，能够有效提高计算效率。

 **说明** 在ServiceRole中添加AliyunOSSReadOnlyAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：

```
{
  "bucket": "my-bucket",
  "object": "my-object"
}
```

- 输出参数：指定文件的MD5值。

logstore-replication：日志备份

您可以通过该函数订阅日志服务SLS的实时数据，并复制数据到另一个Logstore。更多信息，请参见[aliyun-log-fc-functions](#)。

 **说明** 使用该函数前，您需要完成以下操作。

- 开通[日志服务SLS](#)。
- 在ServiceRole中添加AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。
- 由于该函数读取数据并批量写到Logstore，因此建议您为函数设置384 MB以上的内存规格。如果Logstore日志流量较大，建议设置120s以上的超时时间。

- 输入参数：当您增加Logstore的shardcursor等信息后，您可以在function event输入指令，用于指示Function开始、停止读取数据的位置，并为函数自定义逻辑提供额外配置（例如执行数据加工、数据写入哪里等）。当Shard有数据写入时，日志服务SLS会定时触发函数执行。
- 输出参数：执行成功的函数会将源Logstore的数据加工后再写入目标Logstore，并返回一个JSON Object序列化的字符串，包含ingestLines（读取日志行数）、ingestBytes（读取日志字节数）、shipLines（输

出日志行数)和shipBytes(输出日志字节数)。

如果执行过程中发生异常,例如读写Logstore失败,会终止函数运行。

marketplace-weather-demo: 使用云市场服务

您可以通过函数计算使用云市场服务商提供的服务包装。本模板以使用云市场服务商提供的免费天气预报、空气质量、穿衣指数等信息为例介绍该模板函数的使用。



注意 您需要开通云市场里服务商提供的天气预报服务,在[云市场控制台](#)上获取服务的AppCode。

- 输入参数: 输入服务商提供的产品AppCode。
- 输出参数: 以JSON格式传输返回整体天气预报、空气质量等数据。

oss-download-zip-upload: OSS图片下载压缩

您可以通过该函数从Bucket中下载图片,将图片打包成ZIP格式,并把打包后的ZIP文件上传到指定的Bucket中。



说明 在ServiceRole中添加AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作,请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数:
 - EndPoint: 输入Bucket的EndPoint。
 - BucketName: 输入Bucket名称。
- 输出参数:
 - ZipFileSize: 打包后的文件大小,单位为字节。

oss-get-object-info: 获取对象信息

您可以通过该函数获取OSS指定Bucket中Object的详细信息。



说明 在ServiceRole中添加AliyunOSSReadOnlyAccess和AliyunLogFullAccess权限。具体操作,请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数:
 - EndPoint: Bucket的EndPoint。
 - BucketName: Bucket名称。
- 输出参数:
 - last_modified: Object的最后更新时间。
 - key: Object的名称。

oss-shipper-csv: 构建CSV投递OSS

该函数订阅日志服务SLS的实时数据,对于配置的日志字段,按序取出其值构建出一个CSV日志行,最终由一批数据构成目标CSV文件。该CSV文件可以选择直接存入OSS,或者使用Snappy或GZIP为文件做整体压缩后写入OSS。更多信息,请参见[oss-shipper-csv函数说明](#)。

② 说明 使用该函数前，您需要完成以下操作：

- 构建CSV文件需要在内存中暂存本次函数调用所需处理的全部日志数据，请根据日志量为函数设置足够的内存规格。
- 在ServiceRole中添加AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：当您增加Logstore的shardcursor等信息后，您可以在function event输入指令，用于指示Function开始、停止读取数据的位置，并为函数自定义逻辑提供额外配置（例如执行数据加工、数据写入哪里等）。当Shard有数据写入时，日志服务SLS会定时触发函数执行。
- 函数输出：执行成功的函数会将源Logstore数据处理为CSV后，写到OSS。

read-write-tablestore-nodejs12：读写表格存储

当您通过Put Row在表格存储中写入数据时，您可以通过该函数再将该数据通过Get Row读出来。

② 说明 使用该函数前，您需要完成以下操作：

- 开通表格存储并且预先创建好实例和数据表。更多信息，请参见[表格存储概述](#)。
- 在ServiceRole中添加AliyunLogFullAccess和AliyunOTSFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：自定义。
- 输出参数：Get Row的结果。

read-write-tablestore-python3：读写表格存储

当您通过Put Row在表格存储中写入数据时，您可以通过该函数再将该数据通过Get Row读出来。

② 说明 使用该函数前，您需要完成以下操作：

- 开通表格存储并且预先创建好实例和数据表。更多信息，请参见[表格存储概述](#)。
- 在ServiceRole中添加AliyunLogFullAccess和AliyunOTSFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

- 输入参数：自定义。
- 输出参数：Get Row的结果。

return-apigateway-string：API网关返回串

您可以通过该函数获取从API网关传输过来的参数，包括HTML页面、JSON和图片。

- 输入参数：APIPara，在API网关里配置请求参数。
- 输出参数：api_rep，返回API网关标准响应串。

1.2. 使用模板创建函数

函数计算提供了各类函数模板，使用控制台创建函数的过程中，基于函数模板您可以快速创建对应的函数服务，例如网络爬虫、图像自动分类、访问阿里云的OSS服务等。本文以创建一个网络爬虫函数为例介绍如何在函数计算控制台使用模板创建函数。

前提条件

- **创建服务。**
- 由于该函数涉及到OSS的读取和写入，使用前您需要在服务权限中增加AliyunOSSFullAccess和AliyunLogFullAccess权限。具体操作，请参见[授予函数计算访问其他云服务的权限](#)。

操作步骤

1. 登录[函数计算控制台](#)。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏，选择地域。
4. 在**服务列表**页面，单击目标服务。
5. 在**函数管理**页面，单击**创建函数**。
6. 在**创建函数**页面，单击**使用模板创建**，然后在选择函数模板区域，单击image-crawler卡片的任意位置。
7. 在**基本设置**区域，填写函数相关参数，然后单击**创建**。

基本设置

名称

13/64

函数模板

image-crawler

[重新选择模板](#)

实例类型

弹性实例

性能实例

内存规格

512 MB

[手动输入内存大小](#)

参数说明如下。

参数	操作
名称	填写自定义的函数名称。
函数模板	默认为您选择的模板，如果单击 重新选择模板 ，将返回到函数模板选择步骤。
实例类型	选择适合您的实例类型。 ○ 弹性实例 ○ 性能实例
内存规格	设置函数执行内存。 ○ 选择输入：在下拉列表中选择所需内存。 ○ 手动输入：仅适用于弹性实例，单击手动输入内存大小，可自定义函数执行内存。取值范围[128, 3072]，单位为MB。 ? 说明 输入的内存必须为64 MB的倍数。

函数创建完成后，页面跳转至函数详情页面的**函数代码**页签。

8. 在**函数代码**页签，将代码中的<your region>和<your bucket name>替换成您的资源信息，然后单击**保存并部署**。您也可以对其他代码进行在线编辑和改造。

9. 单击  图标，从下拉列表中，选择配置测试参数。在配置测试参数面板，修改event参数，然后单击确定。
10. 单击测试函数，查看函数的执行结果。

2. 基于函数计算和TensorFlow的Serverless AI推理

本文以Python语言为例，介绍如何通过Serverless Devs开发工具，将示例代码部署到函数计算控制台，并安装相关依赖。

背景信息

通过上传一张猫或狗的照片，识别出该照片内的动物是猫还是狗。



关于本示例的详细信息，请参见[示例工程](#)。

前提条件

- 开通函数计算。建议您使用按量付费的方式购买函数计算。详细信息，请参见[开通服务](#)和[按量付费](#)。
- 开通存储服务NAS。关于存储服务NAS的计费信息，请参见[入门概述](#)和[计费方式](#)。
- [安装Serverless Devs和Docker](#)。
- [配置Serverless Devs](#)。

操作步骤

1. 执行以下命令，克隆项目。

```
git clone https://github.com/awesome-fc/cat-dog-classify.git
```

2. 安装依赖。

- i. 执行以下命令进入项目目录。

```
cd cat-dog-classify
```

ii. 执行以下命令安装依赖。

```
s build
```

输出示例：

```
[2021-12-09 07:26:39] [INFO] [S-CLI] - Start ...
[2021-12-09 07:26:40] [INFO] [FC-BUILD] - Build artifact start...
[2021-12-09 07:26:40] [INFO] [FC-BUILD] - Use docker for building.
[2021-12-09 07:26:40] [INFO] [FC-BUILD] - Build function using image: registry.<regionId>.aliyuncs.com/aliyunfc/runtime-python3.6:build-1.9.20
[2021-12-09 07:26:40] [INFO] [FC-BUILD] - begin pulling image registry.<regionId>.aliyuncs.com/aliyunfc/runtime-python3.6:build-1.9.20, you can also use docker pull registry.cn-beijing.aliyuncs.com/aliyunfc/runtime-python3.6:build-1.9.20 to pull image by yourself.
build-1.9.20: Pulling from aliyunfc/runtime-python3.6
f49cf87b52c1: Already exists
.....
01ce50b4eb85: Already exists
02b807385deb: Pull complete
.....
9b9fdb8de506: Pull complete
Digest: sha256:a9a6dab2d6319df741ee135d9749a90b2bb834fd11ee265d1fb106053890****
Status: Downloaded newer image for registry.<regionId>.aliyuncs.com/aliyunfc/runtime-python3.6:build-1.9.20
builder begin to build
[2021-12-09 07:27:57] [INFO] [FC-BUILD] - Build artifact successfully.
Tips for next step
=====
* Invoke Event Function: s local invoke
* Invoke Http Function: s local start
* Deploy Resources: s deploy
End of method: build
```

执行完安装依赖的命令后，Serverless Devs会自动安装相关依赖包，并将第三方库下载到`.s/build/artifacts/cat-dog/classify/.s/python`目录内。

3. 上传依赖到NAS。

当您在安装依赖时，函数计算引用的代码包在解压后可能会出现大于代码包限制的情况，为了减少代码包的体积，您可以将大内存的依赖和相对较大的模型参数文件存放在NAS中。

i. 执行以下命令，初始化NAS。

```
s nas init
```

输出示例：

```
[2021-12-09 07:29:58] [INFO] [S-CLI] - Start ...
[2021-12-09 07:29:59] [INFO] [FC-DEPLOY] - Using region: cn-shenzhen
[2021-12-09 07:29:59] [INFO] [FC-DEPLOY] - Using access alias: default
[2021-12-09 07:29:59] [INFO] [FC-DEPLOY] - Using accessKeyID: LTAI4G4cwJkK4Rza6xd9*
***
[2021-12-09 07:29:59] [INFO] [FC-DEPLOY] - Using accessKeySecret: eCc0GxSpzfqlDVspn
qqd6nmYNN****
.....
[2021-12-09 07:30:01] [INFO] [FC-DEPLOY] - Generated vpcConfig:
securityGroupId: sg-wz90ulsyk2h1f14b****
vSwitchId: vsw-wz9qnuult4q5g4f7n****
vpcId: vpc-wz9x9bzs0wtvjgt6n****
.....
[2021-12-09 07:30:15] [INFO] [FC-DEPLOY] - Checking Trigger httpTrigger exists
Make service _FC_NAS_cat-dog success.
Make function _FC_NAS_cat-dog/nas_dir_checker success.
Make trigger _FC_NAS_cat-dog/nas_dir_checker/httpTrigger success.
[2021-12-09 07:30:25] [INFO] [FC-DEPLOY] - Checking Service _FC_NAS_cat-dog exists
[2021-12-09 07:30:25] [INFO] [FC-DEPLOY] - Checking Function nas_dir_checker exists
[2021-12-09 07:30:26] [INFO] [FC-DEPLOY] - Checking Trigger httpTrigger exists
There is auto config in the service: _FC_NAS_cat-dog
[2021-12-09 07:30:26] [INFO] [FC-DEPLOY] - Generated nasConfig:
groupId: 10003
mountPoints:
- fcDir: /mnt/auto
  nasDir: /cat-dog
  serverAddr: 2bfb748****.cn-shenzhen.nas.aliyuncs.com
userId: 10003
cat-dog:
  userId: 10003
  groupId: 10003
  mountPoints:
  -
    serverAddr: 2bfb748****.cn-shenzhen.nas.aliyuncs.com
    nasDir: /cat-dog
    fcDir: /mnt/auto
```

ii. 执行以下命令，上传依赖到NAS。

```
s nas upload -r .s/build/artifacts/cat-dog/classify/.s/python/ /mnt/auto/python
```

输出示例：

```
[2021-12-09 07:33:14] [INFO] [S-CLI] - Start ...
Packing ...
Package complete.
Upload done
Tips for next step
=====
* Invoke remote function: s invoke
End of method: nas
```

iii. 执行以下命令，上传模型到NAS。

```
s nas upload -r src/model/ /mnt/auto/model
```

输出示例：

```
[2021-12-09 07:52:26] [INFO] [S-CLI] - Start ...
Packing ...
Package complete.
Upload done
Tips for next step
=====
* Invoke remote function: s invoke
End of method: nas
```

iv. 执行以下命令，查看NAS目录。

```
s nas command ls /mnt/auto/
```

输出示例：

```
[2021-12-09 07:53:01] [INFO] [S-CLI] - Start ...
model
python
Tips for next step
=====
* Invoke remote function: s invoke
End of method: nas
```

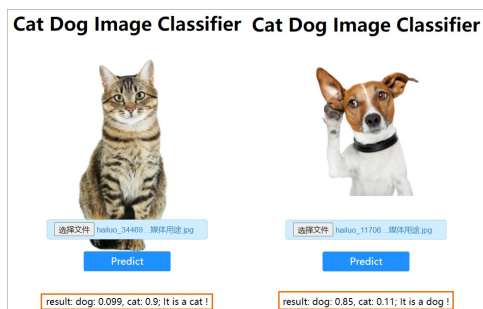
4. 执行以下命令，部署项目。

```
s deploy
```

输出示例：

```
[2021-12-09 07:56:15] [INFO] [S-CLI] - Start ...
[2021-12-09 07:56:16] [INFO] [FC-DEPLOY] - Using region: cn-shenzhen
[2021-12-09 07:56:16] [INFO] [FC-DEPLOY] - Using access alias: default
[2021-12-09 07:56:16] [INFO] [FC-DEPLOY] - Using accessKeyID: LTAI4G4cwJkK4Rza6xd9****
[2021-12-09 07:56:16] [INFO] [FC-DEPLOY] - Using accessKeySecret: eCc0GxSpzfq1DVspnqqd6
nmYNN****
.....
[2021-12-09 07:56:19] [INFO] [FC-DEPLOY] - Generated logConfig:
enableInstanceMetrics: true
enableRequestMetrics: true
logBeginRule: ~
logstore: fc-service-cat-dog-logstore
project: 188077086902****-cn-shenzhen-logproject
.....
There is auto config in the service: cat-dog
Tips for next step
=====
* Display information of the deployed resource: s info
* Display metrics: s metrics
* Display logs: s logs
* Invoke remote function: s invoke
* Remove Service: s remove service
* Remove Function: s remove function
* Remove Trigger: s remove trigger
* Remove CustomDomain: s remove domain
cat-dog:
  region:  cn-shenzhen
  service:
    name: cat-dog
  function:
    name:      classify
    runtime:   python3
    handler:   predict.handler
    memorySize: 1024
    timeout:   120
  url:
    system_url:  https://188077086902****.cn-shenzhen.fc.aliyuncs.com/2016-08-15/prox
y/cat-dog/classify/
    custom_domain:
      -
        domain: http://classify.cat-dog.188077086902****.cn-shenzhen.fc.devsapp.net
  triggers:
    -
      type: http
      name: httpTrigger
```

成功部署该项目后，您可以在执行输出中查看到函数计算生成的临时域名，通过该域名可以访问刚部署的函数，例如您可以使用浏览器访问该域名，然后选择目标图片识别图中的动物：



❓ 说明 临时域名仅用作演示以及开发，具有时效性。如需用作生产，请绑定已经在阿里云备案的域名。详细信息，请参见[配置自定义域名](#)。

使用预留消除冷启动毛刺

函数计算具有动态伸缩的特性，根据并发请求量，自动弹性扩容出执行环境。在这个典型的深度学习示例中，加载依赖和模型参数消耗的时间很长，在您设置的1 GB规格的函数中，并发访问的时间为10s左右，有时可能大于20s。

因此不可避免的会出现函数调用毛刺的情况，即冷启动时间大于10s，在这种情况下，您可以使用设置预留的方式来避免冷启动。您可以在项目目录内执行以下命令消除冷启动毛刺：

```
s provision put --target 10 --qualifier LATEST
```

同时，当您需要了解服务器的最大承受能力，实现更好地运行和开发时，您可以使用Serverless Devs的压测命令对指定的函数进行压测。详细信息，请参见[Serverless Devs操作命令](#)。

🔔 注意 当您完成压测后，请执行以下命令取消预留：

```
s provision put --target 0 --qualifier LATEST
```

更多信息

您可以单击[点我去体验](#)，体验本文介绍的基于函数计算实现AI推理的最佳实践。

[点我去体验](#)