

ALIBABA CLOUD

阿里云

FPGA云服务器 最佳实践

文档版本：20201019

 阿里云

法律声明

阿里云提醒您在使用或阅读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。
<code>Courier</code> 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。
<i>斜体</i>	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 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.FPGA RTL开发流程最佳实践	05
1.1. 使用f1 RTL	05
1.2. f3实例RTL开发最佳实践	07
2.FPGA OpenCL开发流程最佳实践	12
2.1. f1实例OpenCL开发最佳实践	12
2.2. f3实例OpenCL开发最佳实践	16
3.AI推理	23
3.1. FPGA云服务器加速广告CTR预估	23
4.图片转码	24
4.1. FPGA云服务器加速图片和视频转码	24
5.图像处理	26
5.1. 在f1实例中使用CTAccel图像处理（CIP）加速器	26

1.FPGA RTL开发流程最佳实践

1.1. 使用f1 RTL

本文介绍如何使用f1 RTL（Register Transfer Level）。

前提条件

在开始本教程之前，请确保您已完成下列操作：

- 已注册阿里云账号。如还未注册，请先完成[账号注册](#)。
- 已创建f1实例并且实例能访问公网。

 **说明** f1实例只能使用镜像市场的FaaS F1基础镜像。更多详情，请参见[创建f1实例](#)。

- 已在f1实例所在的安全组中添加规则并放行SSH（22）端口。
- 已在[ECS管理控制台](#)f1实例详情页上获取实例ID。
- 已创建OSS Bucket专门用于FaaS服务。

Bucket与f1实例必须属于同一个账号、同一个地域。若尚未创建，请参见[创建一个OSS Bucket](#)。

- 如需加密文件，请先开通密钥管理服务（KMS）。

若尚未开通，请参见[开通密钥管理服务（KMS）](#)。

- 使用RAM用户操作FPGA，必须先完成以下操作：

- 创建RAM用户并授权，详情请参见[创建RAM用户及为RAM用户授权](#)。

您需要为RAM用户授予的权限为：AliyunECSReadOnlyAccess、AliyunOSSFullAccess和AliyunRAMFullAccess。

- 授权FaaS服务角色，授权页面请参见[授权FaaS服务角色](#)。
- 获取AccessKey ID和AccessKey Secret。

背景信息

在开始本教程之前，请阅读以下注意事项：

- 本教程中所有操作都必须由同一个账号在同一地域里执行。
- 强烈建议您使用RAM用户操作FaaS实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。在操作FPGA镜像及下载时，因为您需要从指定的OSS Bucket下载原始DCP工程，所以您必须为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号访问指定的OSS Bucket。如果需要对IP加密，必须授予RAM用户KMS相关权限。如果需要做权限检查，必须授予查看用户资源的权限。

操作步骤

1. 远程连接f1实例。具体操作，请参见[远程连接Linux实例](#)。
2. 运行以下命令配置基础环境。

```
source /opt/dcp1_1/script/f1_env_set.sh
```

3. 依次运行以下命令编译工程。

```
cd /opt/dcp1_1/hw/samples/dma_afu
```

```
afu_synth_setup --source hw/rtl/filelist.txt build_synth
```

```
cd build_synth/
```

```
run.sh
```

 **说明** 编译时间较长，请耐心等待。

4. 制作镜像。

- i. 依次运行以下命令配置 `PATH` 环境变量并为 `faascmd` 文件添加可执行权限。

```
export PATH=$PATH:/opt/dcp1_1/script/
```

```
chmod +x /opt/dcp1_1/script/faascmd
```

- ii. 依次运行以下命令初始化 `faascmd` 配置。

```
# 将hereIsYourSecretId替换为您的AccessKey ID, hereIsYourSecretKey替换为您的AccessKey Secret
```

```
faascmd config --id=hereIsYourSecretId --key=hereIsYourSecretKey
```

```
# 将hereIsYourBucket换为华东1地域里OSS Bucket名称
```

```
faascmd auth --bucket=hereIsYourBucket
```

- iii. 在 `/opt/dcp1_1/hw/samples/dma_afu` 目录下，运行以下命令上传 `gbs` 文件。

```
faascmd upload_object --object=dma_afu.gbs --file=dma_afu.gbs
```

- iv. 运行以下命令制作镜像。

```
# 将hereIsYourImageName替换为您的镜像名称
```

```
faascmd create_image --object=dma_afu.gbs --fpgatype=intel --name=hereIsYourImageName --tags=hereIsYourImageTag --encrypted=false --shell=V1.1
```

5. 下载镜像。

- i. 运行以下命令查看镜像是否制作成功。

```
faascmd list_images
```

返回结果里，如果出现 `"State": "success"`，表示镜像制作成功。请记录返回结果里显示的 `FpgaImageUUID` 的值，稍后会用到。

ii. 运行以下命令获取FPGA ID。

```
# 将herelsYourInstanceId替换为您的f1实例ID
faascmd list_instances --instanceId=herelsYourInstanceId
```

返回结果如下图所示。请记录FpgaUUID的值。

iii. 运行以下命令下载FPGA镜像到f1实例。

```
# 将herelsYourInstanceId替换为刚刚保存的实例ID；将herelsFpgaUUID替换为上一条命令中记下的Fp
gaUUID；将herelsImageUUID替换为上一步记下FpgaImageUUID
faascmd download_image --instanceId=herelsYourInstanceId --fpgauid=herelsFpgaUUID --f
pgatype=intel --imageuid=herelsImageUUID --imagetype=afu --shell=V1.1
```

iv. 运行以下命令检查镜像是否下载成功。

```
# 将herelsYourInstanceId替换为刚刚保存的实例ID；将herelsFpgaUUID替换为上一条命令中记下的Fp
gaUUID
faascmd fpga_status --instanceId=herelsYourInstanceId --fpgauid=herelsFpgaUUID
```

如果返回结果中出现 "TaskStatus":"operating"，且FpgaImageUUID的值和下载镜像时FpgaImageUUID的值一致，说明下载成功。

6. (可选) 如果没有开启Huge pages，运行以下命令启用Huge pages。

```
sudo bash -c "echo 20 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages"
```

7. 依次运行以下命令测试。

```
cd /opt/dcp1_1/hw/samples/dma_afu/sw
```

```
make
```

```
sudo LD_LIBRARY_PATH=/opt/dcp1_1/hw/samples/dma_afu/sw:$LD_LIBRARY_PATH ./fpga_dma_t
est 0
```

输出结果如下图所示，说明测试完成。


1.2. f3实例RTL开发最佳实践

本文描述基于f3实例的RTL (Register Transfer Level) 开发流程。

前提条件

- 已注册阿里云账号。如还未注册，请先完成[账号注册](#)。
- 已创建f3实例，并且实例能访问公网。具体操作，请参见[创建f3实例](#)。

- 已在f3实例所在安全组中添加规则放行SSH（22）端口。具体操作，请参见[添加安全组规则](#)。
- 登录[ECS管理控制台](#)，在f3实例的详情页上，获取实例ID。
- 已在华东2创建一个OSS Bucket，专门用于FaaS服务。具体操作，请参见[创建一个OSS Bucket](#)。

 **说明** 这个Bucket会对FaaS管理账号开通读写权限，因此不建议您存储与FaaS无关的内容。

- 使用RAM用户操作FPGA，必须先完成以下操作：
 - 创建RAM用户并授权，详情请参见[创建RAM用户及为RAM用户授权](#)。
您需要为RAM用户授予的权限为：AliyunECSReadOnlyAccess、AliyunOSSFullAccess和AliyunRAMFullAccess。
 - 授权FaaS服务角色，授权页面请参见[授权FaaS服务角色](#)。
 - 获取AccessKey ID和AccessKey Secret。


背景信息

开始操作之前，您需要了解以下注意事项。

- 本文所述所有操作必须由同一个账号在同一个地域执行。
- 强烈建议您使用RAM用户操作FPGA实例。基于最小授权原则，建议您不要对RAM用户过度授权，而只授予RAM用户刚好满足其工作所需的权限，例如访问OSS bucket获取原始DCP/xclbin文件、上传Vivado编译log、操作指定的ECS实例等。您还需要指定RAM角色AliyunFAASDefaultRole，FaaS服务默认使用此角色来访问您在其他云产品中的资源，其权限策略AliyunFAASRolePolicy还包括KMS相关的权限，以便您使用KMS服务对IP进行加密。

操作步骤

1. [远程连接Linux实例](#)。

 **说明** 编译工程时需要2~3小时。建议您使用nohup或者VNC连接实例，以免编译时意外退出。

2. [下载并解压RTL参考设计](#)。
3. [配置环境](#)。

- 如果驱动为 `xdma`，需要运行以下命令来配置环境。

```
source /root/xbinst_oem/F3_env_setup.sh xdma #每打开一个终端窗口就需要执行该命令一次
```

- 如果驱动为 `xocl`，则需要运行以下命令来配置环境。

```
source /root/xbinst_oem/F3_env_setup.sh xocl #每打开一个终端窗口就需要执行该命令一次
```


 **说明** 配置环境主要包括安装xdma驱动或xocl驱动，设置vivado环境变量，检查vivado license，检测aliyun-f3 sdaccel平台，2018.2 runtime配置和faascmd版本检测。

4. [指定OSS存储空间](#)。


```
faascmd config --id=<herelsYourSecretId> --key=<herelsYourSecretKey> #将<herelsYourSecretId>
和<herelsYourSecretKey>替换为您的RAM用户AK信息
faascmd auth --bucket=herelsYourBucket # 将<herelsYourBucket>替换为您创建的OSS Bucket名称
```

5. 运行以下命令编译RTL工程。

```
cd <您之前解压的路径>/hw/ # 进入解压后的hw路径
sh compiling.sh
```

 **说明** 编译工程需要2~3小时。

6. 上传网表文件，并下载FPGA镜像。您可以采用脚本化流程或者单步操作流程完成该步骤。

- 脚本化流程：仅适用于配备单块FPGA卡的f3实例。
 - a. 运行以下命令上传并生成镜像文件。

```
sh /root/xbinst_oem/tool/faas_upload_and_create_image.sh <需要上传的压缩包文件名>
```

- b. 下载镜像文件。

```
sh /root/xbinst_oem/tool/faas_download_image.sh <压缩包的文件名> <0/1> # 最后的数字<0/
1>为实例中fpga的序号
```

0为FaaS实例中的第一个FPGA，单芯片实例序号一律为0，对多芯片实例，例如4芯片的序号为0, 1, 2, 3。

如果需要对多个FPGA下载同一个镜像，可以在命令的末尾添加相应的序号。例如，对4芯片FPGA下载同一镜像的命令为：

```
sh /root/xbinst_oem/tool/faas_download_image.sh <压缩包的文件名> 0
sh /root/xbinst_oem/tool/faas_download_image.sh <压缩包的文件名> 1
sh /root/xbinst_oem/tool/faas_download_image.sh <压缩包的文件名> 2
sh /root/xbinst_oem/tool/faas_download_image.sh <压缩包的文件名> 3
```

- 单步操作流程：使用faascmd工具进行操作。具体操作，请参见[使用faascmd](#)。
 - a. 依次运行以下命令，将压缩包上传到您个人的OSS Bucket，再将存放在您个人OSS Bucket中的gbs上传到FaaS管理单元的OSS Bucket中。

```
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --name=<herelsFPGAImageName>
e> --tags=<herelsFPGAImageTag> --encrypted=false --shell=<herelsShellVersionOfFPGA>
```

b. 运行命令查看FPGA镜像是否处于可下载状态。

```
faascmd list_images
```

在返回结果中：

- 如果 "State" 为 `compiling` ，表示FPGA镜像处于编译状态，您需要继续等待。
- 如果 "State" 为 `success` ，表示FPGA镜像已经可以下载。您需要找到并记录FpgaImageUUID。

c. 运行以下命令。在命令返回结果中，您需要找到并记录FpgaUUID。

```
faascmd list_instances --instanceId=<hereIsYourInstanceId> # 将<hereIsYourInstanceId>替换为f3实例ID
```

d. 运行以下命令下载FPGA镜像。

```
# 将<hereIsYourInstanceId>替换为f3的实例ID，<hereIsFpgaUUID>替换为您获取的FpgaUUID，<hereIsImageUUID>替换为您获取的FpgaImageUUID  
faascmd download_image --instanceId=<hereIsYourInstanceId> --fpgauuid=<hereIsFpgaUUID> --fpgatype=xilinx --imageuuid=<hereIsImageUUID> --imagetype=afu --shell=<hereIsShellVersionOfFpga>
```

e. 运行以下命令查看镜像是否下载成功。

```
# 将<hereIsFpgaUUID>替换为您获取的FpgaUUID，<hereIsYourInstanceId>替换为f3实例ID。  
faascmd fpga_status --fpgauuid=<hereIsFpgaUUID> --instanceId=<hereIsYourInstanceId>
```

以下为返回结果示例。如果显示的FpgaImageUUID与您获取的FpgaImageUUID一致，并且显示 "TaskStatus":"valid" ，说明镜像下载成功。

FAQ

问题一：上传镜像时出现异常，如何查看异常详情？

如果您的工程在上传生成镜像的过程中出现异常，例如云上编译服务器编译报错，您可以通过以下两种方式来查看异常详情：

- 查看 `faas_compiling.log`。使用上传脚本 `faas_upload_and_create_image.sh` 时，如果编译失败会自动下载并打印 `faas_compiling.log` 到 terminal 中。
- 手动执行命令查看编译 log 文件：

```
sh /root/xbinst_oem/tool/faas_checklog.sh <bit.tar.gz之前上传的压缩包文件名>
```

问题二：如何重新加载镜像？

您可以参考以下步骤重新加载镜像：

1. 卸载驱动。

- 如果您安装了 `xdma` 驱动，则需要在实例中运行 `sudo rmmod xdma` 命令卸载驱动。
- 如果您安装了 `xocl` 驱动，则需要在实例中运行 `sudo rmmod xocl` 命令卸载驱动。

2. 下载镜像。您可以使用以下两种方式之一：

- 使用脚本：

```
sh faas_download_image.sh bit.tar.gz <0/1> #最后的数字为实例中FPGA的序号
```

- 使用 `faascmd`：

```
faascmd download_image --instanceId=hereIsYourInstanceId --fpgauid=hereIsFpgaUUID --fpgaType=xilinx --imageuid=hereIsImageUUID --imageType=afu --shell=hereIsShellVersionOfFpga
```

3. 安装驱动。

- 如果您需要安装 `xdma` 驱动，请运行以下命令。

```
sudo depmod
sudo modprobe xdma
```

- 如果您需要安装 `xocl` 驱动，请运行以下命令。

```
sudo depmod
sudo modprobe xocl
```

2.FPGA OpenCL开发流程最佳实践

2.1. f1实例OpenCL开发最佳实践

本文介绍如何在f1实例上使用OpenCL（Open Computing Language）制作镜像文件，并烧写到FPGA芯片中。

前提条件

- 已注册阿里云账号。如还未注册，请先完成[账号注册](#)。
- 已创建f1实例并且实例能访问公网。

 **说明** f1实例只能使用镜像市场的FaaS F1基础镜像。更多详情，请参见[创建f1实例](#)。

- 已在f1实例所在的安全组中添加规则并放行SSH（22）端口。
- 已在[ECS管理控制台](#)f1实例详情页上获取实例ID。
- 已创建OSS Bucket专门用于FaaS服务。

Bucket与f1实例必须属于同一个账号、同一个地域。若尚未创建，请参见[创建一个OSS Bucket](#)。

- 如需加密文件，请先开通密钥管理服务（KMS）。
若尚未开通，请参见[开通密钥管理服务（KMS）](#)。

- 使用RAM用户操作FPGA，必须先完成以下操作：

- 创建RAM用户并授权，详情请参见[创建RAM用户及为RAM用户授权](#)。

您需要为RAM用户授予的权限为：AliyunECSReadOnlyAccess、AliyunOSSFullAccess和AliyunRAMFullAccess。

- 授权FaaS服务角色，授权页面请参见[授权FaaS服务角色](#)。
- 获取AccessKey ID和AccessKey Secret。

背景信息

开始操作之前，请阅读以下注意事项：

- 本教程中所有操作都必须由同一个账号在同一地域里执行。
- 强烈建议您使用RAM用户操作FaaS实例。为了防止意外操作，您需要让RAM用户仅执行必要的操作。在操作FPGA镜像及下载时，因为您需要从指定的OSS Bucket下载原始DCP工程，所以您必须为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号访问指定的OSS Bucket。如果需要对IP加密，必须授予RAM用户KMS相关权限。如果需要做权限检查，必须授予查看用户资源的权限。

操作步骤

在f1实例上，使用OpenCL Example制作镜像文件，并烧写到FPGA芯片中的操作步骤如下：

1. [步骤一：远程连接实例](#)
2. [步骤二：安装基础环境](#)
3. [步骤三：下载官方的OpenCL Example](#)
4. [步骤四：上传配置文件](#)
5. [步骤五：下载镜像到f1实例](#)

6. 步骤六：将FPGA镜像烧录到FPGA芯片

步骤一：远程连接实例

具体操作，请参见[远程连接Linux实例](#)。

步骤二：安装基础环境

运行以下脚本安装基础环境。

```
source /opt/dcp1_1/script/f1_env_set.sh
```

步骤三：下载官方的OpenCL Example

1. 依次运行以下命令创建并切换到`/opt/tmp`目录。

```
mkdir -p /opt/tmp
```

```
cd /opt/tmp
```

此时，您位于`/opt/tmp`目录下。

进入tmp目录

2. 依次运行以下命令下载并解压Example文件。

```
wget https://www.altera.com/content/dam/altera-www/global/en_US/others/support/examples/download/exm_opencl_matrix_mult_x64_linux.tgz
```

```
tar -zxvf exm_opencl_matrix_mult_x64_linux.tgz
```

解压后的目录如下图所示。

解压Example文件

3. 依次运行以下命令进入`matrix_mult`目录并执行编译命令。

```
cd matrix_mult
```

```
aoc -v -g --report ./device/matrix_mult.cl
```

编译过程可能会持续数个小时，您可以再开一个会话，使用`top`命令监控系统资源占用情况，确定编译状态。

步骤四：上传配置文件

1. 依次运行以下命令初始化faascmd工具。

```
#配置环境变量  
export PATH=$PATH:/opt/dcp1_1/script/
```

```
#为faascmd工具添加可执行权限  
chmod +x /opt/dcp1_1/script/faascmd
```

```
#将命令中的<herelsYourSecretId>换为您的AccessKey ID, <herelsYourSecretKey>替换为您的AccessKey Secret  
faascmd config --id=<herelsYourSecretId> --key=<herelsYourSecretKey>
```

```
#将命令中的<herelsYourBucket>换为您的Bucket名称  
faascmd auth --bucket=<herelsYourBucket>
```

- 依次运行以下命令进入 `matrix_mult/output_files` 目录并上传配置文件。

```
cd matrix_mult/output_files
```

此时, 您应该在 `/opt/tmp/matrix_mult/matrix_mult/output_files` 目录下。

```
faascmd upload_object --object=afu_fit.gbs --file=afu_fit.gbs
```

- 运行以下命令使用 `gbs` 制作FPGA镜像。

```
#将命令中的<herelsYourImageName>换为您的镜像名, 将<herelsYourImageTag>替换为您的镜像标签  
faascmd create_image --object=dma_afu.gbs --fpgatype=intel --name=<herelsYourImageName> -  
-tags=<herelsYourImageTag> --encrypted=false --shell=V1.1
```

- 运行以下命令查看镜像是否制作成功。

```
faascmd list_images
```

若返回结果中显示 `"State":"success"`, 表示镜像制作成功。请记录返回结果中显示的 `FpgaImageUUID`, 稍后会用到。

步骤五：下载镜像到f1实例

- 运行以下命令获取FPGA ID。

```
#将命令中的<herelsYourInstanceId>替换为您的FPGA实例ID  
faascmd list_instances --instanceId=<herelsYourInstanceId>
```

返回结果如下图所示。请记录 `FpgaUUID`。

- 运行以下命令下载镜像到f1实例。

```
#将命令中的<herelsYourInstanceID>替换为刚刚保存的实例ID; 将<herelsFpgaUUID>替换为上一条命令中  
记下的FpgaUUID; 将<herelsImageUUID>替换为上一步记下的FpgaImageUUID  
faascmd download_image --instanceId=<herelsYourInstanceID> --fpgauuid=<herelsFpgaUUID> --  
fpgatype=intel --imageuuid=<herelsImageUUID> --imagetype=afu --shell=V0.11
```

- 运行以下命令检查是否下载成功。

```
# 将命令中的<herelsYourInstanceId>替换为刚刚保存的实例ID；将<herelsFpgaUUID>替换为上一条命令中记下的FpgaUUID
faascmd fpga_status --fpgauuid=<herelsFpgaUUID> --instanceId=<herelsYourInstanceId>
```

若返回结果中显示 "TaskStatus":"operating" ，说明下载成功。



步骤六：将FPGA镜像烧录到FPGA芯片

1. 打开步骤二环境的窗口。如果已关闭，重新执行[步骤二](#)操作。
2. 运行以下命令配置OpenCL的运行环境。

```
sh /opt/dcp1_1/openc1/openc1_bsp/linux64/libexec/setup_permissions.sh
```

3. 运行以下命令返回上级目录。

```
cd ../../
```

此时，您位于/opt/tmp/matrix_mult目录下。

4. 执行编译命令。

```
make
# 输出环境配置
export CL_CONTEXT_COMPILER_MODE_ALTERA=3
cp matrix_mult.aocx ./bin/matrix_mult.aocx
cd bin
host matrix_mult.aocx
```

当您看到如下输出时，说明配置完成。请注意，最后一行必须为 **Verification: PASS** 。

```
[root@iZbpXXXXXZ bin]# ./host matrix_mult.aocx
Matrix sizes:
  A: 2048 x 1024
  B: 1024 x 1024
  C: 2048 x 1024
Initializing OpenCL
Platform: Intel(R) FPGA SDK for OpenCL(TM)
Using 1 device(s)
  skx_fpga_dcp_ddr : SKX DCP FPGA OpenCL BSP (acl0)
Using AOCX: matrix_mult.aocx
Generating input matrices
Launching for device 0 (global size: 1024, 2048)
Time: 40.415 ms
Kernel time (device 0): 40.355 ms
Throughput: 106.27 GFLOPS
Computing reference output
Verifying
Verification: PASS
```


2.2. f3实例OpenCL开发最佳实践

本文介绍如何在f3实例上使用OpenCL（Open Computing Language）制作镜像文件，并烧录到FPGA芯片中。

前提条件

- 已注册阿里云账号。如还未注册，请先完成[账号注册](#)。
- 已创建f3实例并为实例分配公网IP。

若尚未创建，请参见[创建f3实例](#)。

 **说明** f3实例只能使用阿里云共享给您的镜像。

- 已在f3实例所在安全组中添加规则并放行SSH（22）端口。
- 已在ECS控制台f3实例的详情页上，获取实例ID。
- 已创建一个OSS Bucket专门用于FaaS服务。

Bucket与f3实例必须属于同一个账号、同一个地域。若尚未创建，请参见[创建一个OSS Bucket](#)。

- 使用RAM用户操作FPGA，必须先完成以下操作：

- 创建RAM用户并授权，详情请参见[创建RAM用户及为RAM用户授权](#)。

您需要为RAM用户授予的权限为：AliyunECSReadOnlyAccess、AliyunOSSFullAccess和AliyunRAMFullAccess。

- 授权FaaS服务角色，授权页面请参见[授权FaaS服务角色](#)。
- 获取AccessKey ID和AccessKey Secret。

背景信息

开始操作之前，您需要了解以下注意事项。

- 本文所述所有操作都必须由同一个账号在同一地域里执行。
- 建议您使用RAM用户操作FaaS实例。您需要为FaaS管理账号创建一个角色，并授予临时权限，让FaaS管理账号能访问指定的OSS Bucket。
- 本文的示例步骤和命令均基于2018.2版本Sdaccel开发环境。若您使用其他版本Sdaccel开发环境，步骤和命令可能会稍有差异。


操作步骤

在f3实例上，使用OpenCL制作镜像文件，并烧写到FPGA芯片中的操作步骤如下：

1. **步骤一：配置环境**
2. **步骤二：编译二进制文件**
3. **步骤三：检查打包脚本**
4. **步骤四：制作镜像**
5. **步骤五：下载镜像**
6. **步骤六：运行Host程序**

步骤一：配置环境

1. 远程连接f3实例。

 **说明** 后面步骤中的编译工程可能会持续数小时，建议您使用screen或者nohup等方式登录，防止ssh超时退出。

2. 运行以下命令安装Screen。

```
yum install screen -y
```

3. 运行以下命令进入Screen。

```
screen -S f3opencl
```

4. 运行以下命令配置环境。

```
source /root/xbinst_oem/F3_env_setup.sh xocl #每打开一个终端窗口就需要执行该命令一次
```

 **说明**

- 配置环境主要包括安装xocl驱动，设置vivado环境变量，检查vivado license，检测aliyun-f3 sdaccel平台，2018.2 runtime配置和faascmd版本检测。
- 如果您要运行sdaccel的仿真，请勿运行以上命令配置环境。您只需要单独配置vivado的环境变量即可。
- 推荐您使用Makefile方式仿真。

步骤二：编译二进制文件

完成以下步骤，编译vadd二进制文件和kernel_global_bandwidth二进制文件：

- 示例一：编译vadd二进制文件

- i. 复制example目录。

```
cp -rf /opt/Xilinx/SDx/2018.2/examples ./
```

- ii. 进入vadd目录。

```
cd examples/vadd/
```

- iii. 运行命令`cat sdaccel.mk | grep "XDEVICE="`查看XDEVICE的值，确保其配置为 `XDEVICE=xilinx_aliyun-f3_dynamic_5_0`。

- iv. 按以下步骤修改common.mk文件。

- a. 运行`vim ../common/common.mk`命令打开该文件。

- b. 在第 61行代码（参数可能在 60~62 行，视您的文件而定）的末尾添加编译参数`--xp param:compiler.acceleratorBinaryContent=dcg`，修改后的代码如下：

```
CLCC_OPT += $(CLCC_OPT_LEVEL) ${DEVICE_REPO_OPT} --platform ${XDEVICE} ${KERNEL_DEFINITION} ${KERNEL_INCLUDES} --xp param:compiler.acceleratorBinaryContent=dcg
```

 说明 由于您必须向编译服务器提交DCP文件，所以需要添加`--xp param:compiler.acceleratorBinaryContent=dcg`编译参数，使得Xilinx® OpenCL™ Compiler (xocc) 编译生成一个布局布线后的DCP文件，而不是bit文件。

- v. 运行以下命令编译程序。

```
make -f sdaccel.mk xbin_hw
```

如果您看到如下界面，说明二进制文件编译已经开始。编译过程可能会持续数个小时，请您耐心等待。




- 示例二：编译kernel_global_bandwidth二进制文件

- i. 依次运行以下命令克隆 `xilinx 2018.2 example`。

```
git clone https://github.com/Xilinx/SDAccel_Examples.git
```

```
cd SDAccel_Examples/
```

```
git checkout 2018.2
```

 说明 git分支必须为2018.2版本。

- ii. 运行`cd getting_started/kernel_to_gmem/kernel_global_bandwidth/`命令进入目录。

- iii. 按以下步骤修改Makefile文件。

- a. 运行`vim Makefile`命令打开该文件。

- b. 设置 `DEVICES=xilinx_aliyun-f3_dynamic_5_0` 。
- c. 在第33行代码中添加编译参数`--xp param:compiler.acceleratorBinaryContent=dcp`，修改后的代码如下：

```
CLFLAGS +=--xp "param:compiler.acceleratorBinaryContent=dcp" --xp "param:compiler.preserveHlsOutput=1" --xp "param:compiler.generateExtraRunData=true" --max_memory_ports bandwidth -DNDDR_BANKS=$(ddr_banks)
```

- iv. 运行以下命令编译程序。

```
make TARGET=hw
```

如果您看到该界面，说明二进制文件编译已经开始。编译工程可能会持续数小时，请您耐心等待。



步骤三：检查打包脚本

运行以下命令检查打包脚本是否存在。

```
file /root/xbinst_oem/sdaccel_package.sh
```

如果返回结果中包含 `cannot open (No such file or directory)`，说明不存在该文件，您需要运行以下命令手动下载打包脚本。

```
wget http://fpga-tools.oss-cn-shanghai.aliyuncs.com/sdaccel_package.sh
```

步骤四：制作镜像

1. 初始化faascmd工具并配置OSS环境。

- i. 运行以下命令设置RAM账号的AccessKey信息。

```
#将<hereIsYourSecretId>和<hereIsYourSecretKey>替换为您的RAM用户AK信息  
faascmd config --id=<hereIsYourSecretId> --key=<hereIsYourSecretKey>
```

- ii. 运行以下命令设置FaaS服务要使用的OSS Bucket。

```
#将<hereIsYourBucket>替换为您创建的OSS Bucket名称  
faascmd auth --bucket=<hereIsYourBucket>
```

2. 运行ls，获取后缀为 `.xclbin` 的文件名。



3. 运行以下命令打包二进制文件。

```
/root/xbinst_oem/sdaccel_package.sh -xclbin=/opt/Xilinx/SDx/2018.2/examples/vadd/bin_vadd_hw.xclbin
```

打包完成后，您会在同一目录下看到一个打包好的文件，如下图所示。



步骤五：下载镜像

您可以采用脚本化流程或者单步操作流程来上传网表文件，并下载FPGA镜像。

- 脚本化流程：仅适用于配备单块FPGA卡的f3实例。


i. 运行以下命令上传并生成镜像文件。

```
sh /root/xbinst_oem/tool/faas_upload_and_create_image.sh <bit.tar.gz需要上传的压缩包文件名>
```



ii. 运行以下命令下载镜像文件。

```
sh /root/xbinst_oem/tool/faas_download_image.sh <bit.tar.gz压缩包的文件名> <0/1> #最后的数字<0/1>为实例中fpga的序号
```

 **说明** 0为FaaS实例中的第一个FPGA。单芯片实例序号一律为0；多芯片实例，例如，4芯片实例的序号为0, 1, 2, 3。

如果需要对多个FPGA下载同一个镜像，可以在命令的末尾添加相应的序号。例如，对4芯片FPGA下载同一镜像的命令如下：

```
sh /root/xbinst_oem/tool/faas_download_image.sh <bit.tar.gz压缩包的文件名> 0
```

```
sh /root/xbinst_oem/tool/faas_download_image.sh <bit.tar.gz压缩包的文件名> 1
```

```
sh /root/xbinst_oem/tool/faas_download_image.sh <bit.tar.gz压缩包的文件名> 2
```

```
sh /root/xbinst_oem/tool/faas_download_image.sh <bit.tar.gz压缩包的文件名> 3
```

- 单步操作流程：使用faascmd工具操作。工具详情，请参见[使用faascmd工具](#)。

i. 依次运行以下命令，将压缩包上传到您个人的OSS Bucket，再将存放在您个人OSS Bucket中的gbs上传到FaaS管理单元的OSS Bucket中。

```
faascmd upload_object --object=bit.tar.gz --file=bit.tar.gz
```

```
faascmd create_image --object=bit.tar.gz --fpgatype=xilinx --name=<hereIsFPGAImageName> --tags=<hereIsFPGAImageTag> --encrypted=false --shell=<hereIsShellVersionOfFPGA>
```



ii. 运行以下命令查看FPGA镜像是否处于可下载状态。

```
faascmd list_images
```

返回结果中：

- 如果FPGA镜像的 "State" 为 "compiling" ，表示FPGA镜像处于编译状态，您需要继续等待。
- 如果FPGA镜像的 "State" 为 "success" ，表示FPGA镜像已可以下载，您需要找到并记录FpgaImageUUID。

iii. 运行以下命令在返回结果中，找到并记录FpgaUUID。

```
faascmd list_instances --instanceId=<hereIsYourInstanceId> #将<hereIsYourInstanceId>替换为f3实例ID
```

iv. 运行以下命令下载FPGA镜像。

```
#将<hereIsYourInstanceId>替换为f3的实例ID，<hereIsFpgaUUID>替换为您获取的FpgaUUID，<hereIsImageUUID>替换为您获取的FpgaImageUUID
faascmd download_image --instanceId=<hereIsYourInstanceId> --fpgauuid=<hereIsFpgaUUID>
--fpgatype=xilinx --imageuuid=<hereIsImageUUID> --imagetype=afu --shell=<hereIsShellVersionOfFpga>
```

v. 运行以下命令查看镜像是否下载成功。

```
faascmd fpga_status --fpgauuid=<hereIsFpgaUUID> --instanceId=<hereIsYourInstanceId> #将<hereIsFpgaUUID>替换为您获取的FpgaUUID，<hereIsYourInstanceId>替换为f3实例ID。
```

以下为返回结果示例。如果显示的FpgaImageUUID与您获取的FpgaImageUUID一致，并且显示 "TaskStatus":"valid" ，说明镜像下载成功。

步骤六：运行Host程序

1. 运行以下命令配置环境。

```
source /root/xbinst_oem/F3_env_setup.sh xocl #每打开一个终端窗口就需要执行该命令一次
```

2. 配置 *sdaccel.ini* 文件。

在Host二进制文件所在目录下，运行 `vim sdaccel.ini` 命令创建 *sdaccel.ini* 文件并输入下列内容。

```
[Debug]
profile=true
[Runtime]
runtime_log = "run.log"
hal_log = hal.log
ert=false
kds=false
```

3. 运行host。

- vadd运行命令如下：

```
make -f sdaccel.mk host
```

```
./vadd bin_vadd_hw.xclbin
```

- kernel_global_bandwidth运行命令如下：

```
./kernel_global
```

如果返回结果中出现 `Test Passed` ，说明测试通过。

其他操作

这里介绍FPGA实例部分常用的操作。

任务	命令
查看帮助文档	<code>make -f ./sdaccel.mk help</code>
软件仿真	<code>make -f ./sdaccel.mk run_cpu_em</code>
硬件仿真	<code>make -f ./sdaccel.mk run_hw_em</code>
只编译host代码	<code>make -f ./sdaccel.mk host</code>
编译生成可以下载的文件	<code>make -f sdaccel.mk xbin_hw</code>
清理工作目录	<code>make -f sdaccel.mk clean</code>
强力清除工作目录	<code>make -f sdaccel.mk cleanall</code>

② 说明

- 仿真时只需要按照Xilinx标准流程操作，不需要配置F3_env_setup环境。
- SDAccel runtime和SDAccel开发平台已在阿里云f3官方镜像中提供。您也可以点击后面的链接直接下载SDAccel runtime和SDAccel开发平台。

3.AI推理

3.1. FPGA云服务器加速广告CTR预估

本实践方案基于阿里云FPGA计算型实例，完整演示加速广告点击率预估（Click-Through Rate）的场景，帮助您快速理解阿里云FPGA云服务器。

解决的问题

- 加速广告点击率预估的计算过程。
- 降低计算成本。

部署架构图



选用的产品

- FPGA云服务器

FPGA云服务器是一款提供了现场可编程门阵列（FPGA）的计算实例，基于阿里云弹性计算框架，用户可以以几分钟内轻松创建FPGA实例，创建自定义的专用硬件加速器。由于FPGA硬件的可重配特性，用户可以对已创建的FPGA硬件加速应用，进行快速擦写和重配，达到低时延硬件与弹性伸缩最好的结合。

更多关于FPGA云服务器的介绍，请参见[FPGA云服务器详情页](#)。

- 云服务器ECS

云服务器（Elastic Compute Service）是阿里云提供的性能卓越、稳定可靠、弹性扩展的IaaS（Infrastructure as a Service）级别云计算服务。云服务器ECS免去了您采购IT硬件的前期准备，让您像使用水、电、天然气等公共资源一样便捷、高效地使用服务器，实现计算资源的即开即用和弹性伸缩。阿里云ECS持续提供创新型服务器，解决多种业务需求，助力您的业务发展。

更多关于云服务器ECS的介绍，请参见[云服务器ECS产品详情页](#)。

- 专有网络VPC

专有网络VPC帮助您基于阿里云构建出一个隔离的网络环境，并可以自定义IP地址范围、网段、路由表和网关等；此外，也可以通过专线、VPN、GRE等连接方式实现云上VPC与传统IDC的互联，构建混合云业务。

更多关于专有网络VPC的介绍，请参见[专有网络VPC产品详情页](#)。

- 对象存储OSS

阿里云对象存储服务（Object Storage Service），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。其数据设计持久性不低于99.999999999%（12个9），服务设计可用性（或业务连续性）不低于99.995%。

更多关于对象存储OSS的介绍，请参见[对象存储OSS产品详情页](#)。

详细信息

[点击查看最佳实践详情](#)

更多最佳实践

[点击查看更多阿里云最佳实践](#)

4. 图片转码

4.1. FPGA云服务器加速图片和视频转码

本实践基于阿里云FPGA计算型实例，完整演示加速视频转码和图片转码两个场景，帮助您快速理解阿里云FPGA云服务器。

场景描述

两个场景中的实现效果如下：

- 加速视频转码：视频编码IP的编译、镜像下载到FPGA以及将MP4视频文件编码为H.265的全过程。
- 加速图片转码：使用基于FPGA的图像格式装换程序，将JPEG格式的图片转换为WebP格式，并进行缩放。

解决的问题

- 提供高画质低码率的实时转码技术方案。
- 提高图片和视频转码效率。
- 降低传输带宽、转码和存储成本。

部署架构图



选用的产品

- **FPGA云服务器**

FPGA云服务器是一款提供了现场可编程门阵列（FPGA）的计算实例，基于阿里云弹性计算框架，用户可以几分钟内轻松创建FPGA实例，创建自定义的专用硬件加速器。由于FPGA硬件的可重配特性，用户对已创建的FPGA硬件加速应用，进行快速擦写和重配，达到低时延硬件与弹性伸缩最好的结合。

更多关于FPGA云服务器的介绍，请参见[FPGA云服务器详情页](#)。

- **云服务器ECS**

云服务器（Elastic Compute Service）是阿里云提供的性能卓越、稳定可靠、弹性扩展的IaaS（Infrastructure as a Service）级别云计算服务。云服务器ECS免去了您采购IT硬件的前期准备，让您像使用水、电、天然气等公共资源一样便捷、高效地使用服务器，实现计算资源的即开即用和弹性伸缩。阿里云ECS持续提供创新型服务器，解决多种业务需求，助力您的业务发展。

更多关于云服务器ECS的介绍，请参见[云服务器ECS产品详情页](#)。

- **专有网络VPC**

专有网络VPC帮助您基于阿里云构建出一个隔离的网络环境，并可以自定义IP地址范围、网段、路由表和网关等；此外，也可以通过专线、VPN、GRE等连接方式实现云上VPC与传统IDC的互联，构建混合云业务。

更多关于专有网络VPC的介绍，请参见[专有网络VPC产品详情页](#)。

- **对象存储OSS**

阿里云对象存储服务（Object Storage Service），是阿里云提供的海量、安全、低成本、高可靠的云存储服务。其数据设计持久性不低于99.999999999%（12个9），服务设计可用性（或业务连续性）不低于99.995%。

更多关于对象存储OSS的介绍，请参见[对象存储OSS产品详情页](#)。

详细信息

[点击查看最佳实践详情](#)

更多最佳实践

[点击查看更多阿里云最佳实践](#)

5. 图像处理

5.1. 在f1实例中使用CTAccel图像处理（CIP）加速器

如果您的某些业务要求ECS实例内不能使用AccessKey，您可以为ECS实例配置RAM角色，ECS实例中部署的应用需要通过实例元数据获取临时授权Token，才可以访问指定资源。本教程介绍如何以无AccessKey的方式访问f1实例资源，从而便捷地为f1实例加载CTAccel CIP加速器。

前提条件

- 已注册阿里云账号。如还未注册，请先完成[账号注册](#)。
- 已完成实名认证。如还未认证，请先完成[实名认证](#)。
- 联系服务商完成以下准备工作，联系方式请参见[CTAccel CIP加速器服务商](#)。
 - 获取开发套件包。
 - 了解支持的镜像类型，并为ECS实例准备自定义镜像。
 - 关联ECS实例镜像和FPGA镜像。

 **说明** ECS实例镜像必须为自定义镜像，申请关联镜像时需提供镜像ID。

- 获取FPGA镜像的FPGAImageUUID。

背景信息

CTAccel图像处理（CIP）加速器是一款基于FPGA的图像处理加速解决方案，可通过将计算负载从CPU转移至FPGA，显著提高图像处理及图像分析的性能。

操作步骤

- **步骤一：创建RAM角色并授予权限**
- **步骤二：创建f1实例并配置RAM角色和标签**
- **步骤二：为已有f1实例配置RAM角色和标签**
- **步骤三：安装faascmd**
- **步骤四：配置CTAccel环境**

步骤一：创建RAM角色并授予权限

创建RAM角色并配置RAM权限用于实现无AccessKey访问。更多实例RAM角色说明，请参见[实例RAM角色概述](#)。

1. 云账号登录[RAM控制台](#)。
2. 新建权限策略。
 - i. 单击[权限管理](#) > [权限策略管理](#)。
 - ii. 单击[新建权限策略](#)。
 - iii. 填写策略名称。本示例中使用faasEcsRamPolicy。
 - iv. 配置模式选择脚本配置。

- v. 填入权限策略内容。以下权限策略用于允许受信服务查看绑定对应标签的ECS实例资源。您可以自定义标签的键和标签值，本示例中使用faas和image。

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": "ecs:DescribeInstances",
      "Effect": "Allow",
      "Resource": "acs:ecs:*:*:*",
      "Condition": {
        "StringEquals": {
          "ecs:tag/faas": "image"
        }
      }
    }
  ]
}
```

- vi. 单击确定。

3. 创建RAM角色。如果已存在AliyunFAASDefaultRole，您可以在已有基础上增加信任策略的内容并授予权限。

 **注意** 如果其它子账号也使用了AliyunFAASDefaultRole，请谨慎修改。

- i. 单击RAM角色管理。
 - ii. 单击新建RAM角色。
 - iii. 类型选择阿里云服务。
 - iv. 填写角色名称，必须是AliyunFAASDefaultRole。
 - v. 受信服务选择云服务器。
 - vi. 单击完成。
 - vii. 单击关闭。
4. 修改RAM角色的信任策略。
 - i. 找到创建的AliyunFAASDefaultRole，在RAM角色名称列中，单击名称。
 - ii. 单击信任策略管理。
 - iii. 单击修改信任策略。

- iv. 按以下信任策略修改，然后单击确定。以下信任策略用于允许ECS服务和FaaS服务扮演该RAM角色。

```
{
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ecs.aliyuncs.com",
          "faas.aliyuncs.com"
        ]
      }
    }
  ],
  "Version": "1"
}
```

5. 授予RAM角色权限。
 - i. 单击权限管理。
 - ii. 单击精确授权。
 - iii. 单击自定义策略。
 - iv. 输入策略名称。例如本示例中的faasEcsRamPolicy。
 - v. 单击确定。
 - vi. 单击关闭。

授权后，ECS服务和FaaS服务可以通过扮演RAM角色AliyunFAASDefaultRole，获取查看绑定标签 `faas:image` 的ECS实例资源的权限，无需再提供AccessKey。

步骤二：创建f1实例并配置RAM角色和标签

请参见以下步骤创建一台f1实例，并在创建过程中配置RAM角色和标签。

1. 登录[ECS管理控制台](#)。
2. 在左侧导航栏，单击实例与镜像 > 实例。
3. 单击创建实例。
4. 完成实例创建。下表中只列出需要重点关注的配置项，请视需要完成其它配置，更多说明请参见[使用向导创建实例](#)。

配置页面	配置项	说明
------	-----	----

配置页面	配置项	说明
基本配置	实例	<ul style="list-style-type: none"> 如果处理jpeg和webp格式的图片，建议选择ecs.f1-c8f1.2xlarge。 如果处理heif格式的图片，建议选择ecs.f1-c28f1.7xlarge。 您可以前往 ECS实例可购买地域 ，查看实例在各地域的可购情况。
	镜像	选择您创建好的自定义镜像。
系统配置	实例RAM角色	AliyunFAASDefaultRole
分组设置	标签	权限策略中定义的标签键和标签值，本示例中使用 <code>faas:image</code> 。

- 在实例列表中，找到f1实例，然后在实例ID/名称列中，单击实例ID。确认已成功为f1实例授予RAM角色并绑定标签。

步骤二：为已有f1实例配置RAM角色和标签

如果您已经创建了f1实例，且f1实例的实例规格和镜像满足要求，请参见以下步骤配置RAM角色和标签。

- 登录[ECS管理控制台](#)。
- 在左侧导航栏，单击实例与镜像 > 实例。
- 找到待操作的f1实例。
- 为f1实例授权RAM角色。
 - 在实例列表中，找到待操作f1实例，然后单击更多 > 实例设置 > 授予/收回RAM角色。
 - 选择RAM角色AliyunFAASDefaultRole，然后单击确定。
- 为f1实例绑定标签。
 - 在实例列表中，找到待操作f1实例，然后单击更多 > 实例设置 > 编辑标签。
 - 单击新建标签，输入上文信任策略中定义的标签键和标签值，然后单击确定。
 - 标签绑定完毕后，单击确定。
- 在实例列表中，找到f1实例，然后在实例ID/名称列中，单击实例ID。确认已成功为f1实例授予RAM角色并绑定标签。

步骤三：安装faascmd

faascmd是阿里云FPGA云服务器（FaaS平台）提供的命令行工具，是基于阿里云Python SDK开发的脚本，您可以通过faascmd命令方便地操作FaaS平台的资源。

- 远程连接Linux实例。
- 检查是否安装了Python，且确保Python版本号必须为2.7.x。

```
python -V
```

返回Python版本号表示已安装Python。如果版本过低，请先升级至2.7.x。如果版本过高，请使用2.7.x版本，否则可能存在语法兼容问题。

3. 安装Python模块。

```
pip install oss2
pip install aliyun-python-sdk-core
pip install aliyun-python-sdk-faas
pip install aliyun-python-sdk-ram
```

4. 检查aliyun-python-sdk-core版本号，确保版本号为2.11.0或以上版本。

```
pip list | grep aliyun-python-sdk-core
```

如果版本过低，请先升级至2.11.0或以上版本。

5. 下载faascmd。

```
curl -o /bin/faascmd http://fpga-tools.oss-cn-shanghai.aliyuncs.com/faascmd-sts
```

6. 为faascmd添加可执行权限。

```
chmod +x /bin/faascmd
```

步骤四：配置CTAccel环境

您可以从服务商提供的开发套件包中获取OPAE和CIP SDK安装包。

1. 安装图片lib库。

```
yum install libjpeg-turbo-* libpng-* libwebp-* libtiff-* jasper jasper-devel
```

2. 切换至OPAE安装包目录，运行以下命令安装OPAE。

```
yum install opae-*
```

3. 切换至CIP SDK安装包目录，运行以下命令安装CIP SDK。

```
yum install accel-*
```

4. 下载初始化配置脚本。

```
curl -o ctaccel-aliyun-init.sh https://fpga-tools.oss-cn-shanghai.aliyuncs.com/ctaccel-aliyun-init.sh
```

5. 加载您从服务商处获取的FPGA镜像，并初始化软件环境。


```
sh ctaccel-aliyun-init.sh <yourFPGAImageUUID>
```

如果出现报错 `HTTP Status: 404 Error:IMAGE NOT FOUND The fpga image you specify is not found!` , 因为未关联ECS镜像和FPGA镜像。请联系服务商进行关联操作。其它faascmd报错的处理方法, 请参见[faascmd工具FAQ](#)。

6. 查看acceld service状态。

```
service acceld status
```

配置成功后, acceld service需要为running状态。

 **说明** 请记录服务启动日期用于查看日志文件。

7. 查看日志文件状态。

```
cat /var/log/acceld-<年-月-日>.log
```

配置成功后, 日志文件中需要正确读取版本号。图中版本号仅为示例, 版本号应和购买时服务商提供的版本一致。

相关文档

- [AttachInstanceRamRole](#)
- [TagResources](#)