

阿里云 云命令行 教程

文档版本：20190905

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 |
|---|-----------------------------------|--|
|  | 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。 |  禁止： 重置操作将丢失用户配置数据。 |
|  | 该类警示信息可能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，恢复业务所需时间约10分钟。 |
|  | 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。 | 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。 | 单击 确定 。 |
| <code>courier</code> 字体 | 命令。 | 执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。 |
| <code>##</code> | 表示参数、变量。 | <code>bae log list --instanceid</code> <code>Instance_ID</code> |
| <code>[]</code> 或者 <code>[a b]</code> | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| <code>{ }</code> 或者 <code>{a b}</code> | 表示必选项，至多选择一个。 | <code>swich {stand slave}</code> |

目录

| | |
|---|----|
| 法律声明..... | I |
| 通用约定..... | I |
| 1 云命令行工具集..... | 1 |
| 1.1 使用Alibaba Cloud CLI管理Serverless Kubernetes集群..... | 1 |
| 1.2 通过Terraform部署Kubernetes托管版集群..... | 3 |
| 1.3 使用kubectl管理Kubernetes集群..... | 6 |
| 1.4 在Alibaba Cloud Shell中快速体验Fun/fcli工具..... | 8 |
| 2 云资源管控..... | 12 |
| 2.1 使用Alibaba Cloud CLI管理弹性容器ECI实例..... | 12 |
| 2.2 使用Packer创建ECS自定义镜像..... | 14 |
| 2.3 复制ECS安全组..... | 16 |
| 3 解决方案..... | 19 |
| 3.1 基于Terraform部署Kubernetes托管版集群及Wordpress应用..... | 19 |
| 3.2 基于弹性容器实例部署Parse Server应用..... | 22 |
| 3.3 通过Helm部署Jenkins..... | 25 |
| 3.4 快速部署智能相册解决方案应用..... | 27 |
| 3.5 快速体验趣视频解决方案..... | 29 |

1 云命令行工具集

1.1 使用Alibaba Cloud CLI管理Serverless Kubernetes集群

您可以通过使用Alibaba Cloud CLI自动创建一个Serverless Kubernetes集群，并获取该集群信息。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-serverless-kubernetes-cli）中集成。您可以通过点击代码块右上角的试用按钮，快速体验使用Alibaba Cloud CLI管理您的Serverless Kubernetes集群，包括创建集群、获取集群信息等操作。

完成本教程后，默认创建以下资源：

- NAT网关
 - 实例数量：1
 - 付费模式：按量付费



说明：

具体计费信息，参见[NAT网关按量计费](#)。

使用限制

在开始使用本教程之前，确保您已经了解以下限制并满足相关要求：

- 确保您已开通如下服务：
 - [容器服务](#)
 - [访问控制（RAM）服务](#)
- 目前阿里云Serverless Kubernetes在公测中，您可以申请[参与公测](#)，免费体验Serverless Kubernetes。
- 请保证您的账户有100元的余额并通过实名认证。
- 您的每个账号默认可以创建的云资源有一定的配额，如果超过配额，集群创建失败。如果您需要提高配额，请提交工单申请。
 - 每个账号默认最多可以创建100个安全组。
 - 每个账号默认最多可以创建60个按量付费的负载均衡实例。
 - 每个账号默认最多可以创建20个EIP。

创建Serverless Kubernetes集群

执行以下命令直接创建Serverless Kubernetes集群：

```
aliyun cs POST /clusters --header "Content-Type=application/json" --body "$(cat create.json)"
```

其中，创建Serverless Kubernetes集群的参数保存在create.json文件中。代码如下：

```
{
  "cluster_type": "Ask",
  "name": "example-serverless-kubernetes",
  "region_id": "cn-shanghai",
  "private_zone": false,
  "vpc_id": "",
  "vswitch_id": ""
}
```

您可以通过以下参数说明自定义您的Serverless Kubernetes集群：

- cluster_type** 集群类型，Serverless Kubernetes集群类型为 "Ask"。
- name** 集群名称，集群名称可以使用大小写英文字母、中文、数字、中划线。
- region_id** 集群所在地域ID。
- private_zone** 是否开启privateZone。
- vpc_id** VPC ID，可空。如果不设置，系统会自动创建VPC，系统创建的VPC网段为192.168.0.0/16。并在VPC中自动创建NAT网关以及配置SNAT规则。



说明：

vpcid和vswitchid只能同时为空或者同时都设置对应的值。

- vswitch_id** 交换机ID，可空。若不设置，系统会自动创建交换机，系统自动创建的交换机网段为192.168.0.0/16。

创建成功后，控制台显示如下输出：

```
{
  "cluster_id": "c61cf530524474386a7a*****",
  "request_id": "348D4C9C-9105-4A1B-A86E-*****",
  "task_id": "T-5ad724ab94a2b109e*****"
}
```

其中，cluster_id表示您创建的集群的唯一id。

您现在可以登录[容器服务控制台](#)，查看通过Alibaba Cloud CLI创建的Serverless Kubernetes集群。

获取Serverless Kubernetes集群信息

执行以下命令查询您已经创建的集群信息。

```
aliyun cs GET /clusters/<cluster_id>
```



说明:

其中, <cluster_id>替换为您需要查询的集群 id。

执行完成后, 控制台显示如下输出:

```
{
  "agent_version": "string",
  "cluster_id": "string",
  "created": "datetime",
  "external_loadbalancer_id": "string",
  "master_url": "string",
  "name": "string",
  "network_mode": "string",
  "region_id": "string",
  "security_group_id": "string",
  "size": "numbers",
  "state": "string",
  "updated": "datetime",
  "vpc_id": "string",
  "vswitch_id": "string",
  ...
}
```

1.2 通过Terraform部署Kubernetes托管版集群

您可以通过Terraform部署一个阿里云容器服务的Kubernetes托管版集群。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录 (tutorial-terraform-managed-kubernetes) 中集成。您可以通过点击代码块右上角的试用按钮, 快速体验通过Terraform自动创建、编排和管理Kubernetes集群。

本教程以如下Terraform模板为例。

```
// Instance_types data source for instance_type
data "alicloud_instance_types" "default" {
  cpu_core_count = "${var.cpu_core_count}"
  memory_size    = "${var.memory_size}"
}

// Zones data source for availability_zone
data "alicloud_zones" "default" {
  available_instance_type = "${var.worker_instance_type == "" ? data
.alicloud_instance_types.default.instance_types.0.id : var.worker_ins
tance_type}"
}
```

```
resource "alicloud_cs_managed_kubernetes" "k8s" {
  name = "${var.k8s_name_prefix == "" ? format("%s-%s", var.example_name, format(var.number_format, count.index+1)) : format("%s-%s", var.k8s_name_prefix, format(var.number_format, count.index+1))}"
  availability_zone = "${lookup(data.alicloud_zones.default.zones[
count.index:length(data.alicloud_zones.default.zones)], "id")}"
  new_nat_gateway = true
  worker_instance_types = ["${var.worker_instance_type == "" ? data.
alicloud_instance_types.default.instance_types.0.id : var.worker_ins
tance_type}"]
  worker_numbers = ["${var.k8s_worker_number}"]
  worker_disk_category = "${var.worker_disk_category}"
  worker_disk_size = "${var.worker_disk_size}"
  password = "${var.ecs_password}"
  pod_cidr = "${var.k8s_pod_cidr}"
  service_cidr = "${var.k8s_service_cidr}"
  install_cloud_monitor = true
  kube_config = "~/.kube/config"
}
```

完成本教程后，您会创建以下资源。其中，容器服务没有任何附加费用，您只需要支付所使用资源（云服务器、负载均衡等）的费用。

- Worker 实例 (ECS)
 - 实例规格: `ecs.n2.medium`
 - 实例数量: 3
 - 系统盘: 20G 高效云盘
- 负载均衡
 - 实例数量: 2
 - 付费模式: 按量付费
- 弹性公网 IP
 - 实例数量: 1
 - 付费模式: 使用流量计费
- NAT 网关
 - 实例数量: 1
 - 付费模式: 按量付费

具体计费信息，参见 [ECS 计费概述](#)、[负载均衡按量计费](#)、[弹性公网 IP 按量计费](#)、[NAT 网关按量计费](#)。

使用限制

在开始使用本教程前，确保您已经了解以下限制并满足相关要求：

- 保证您的账户有100元的余额并通过实名认证，否则无法创建按量付费的ECS实例和负载均衡。

- 随集群一同创建的负载均衡实例只支持按量付费的方式。
- Kubernetes集群仅支持专有网络VPC。
- 您的每个账号默认可以创建的云资源有一定的配额，如果超过配额，集群创建失败。如果您需要提高配额，请提交工单申请。
 - 每个账号默认最多可以创建100个安全组。
 - 每个账号默认最多可以创建60个按量付费的负载均衡实例。
 - 每个账号默认最多可以创建20个EIP。
- 在开始之前，确保您已开通了以下云服务：
 - [容器服务](#)
 - [资源编排](#)
 - [弹性伸缩](#)
 - [访问控制](#)

创建托管版Kubernetes集群

完成以下操作，创建Kubernetes集群：

1. 执行以下命令定位到用来创建Kubernetes集群的Terraform模板的目录。

```
cd ~/terraform-kubernetes-wordpress/kubernetes
```

2. 执行init命令加载[Alibaba Cloud Providers](#)。

```
terraform init
```

3. 执行以下命令部署集群。

```
terraform apply
```



说明：

如果出现 `ErrManagedKubernetesRoleNotAttach` 的错误，请检查所需服务是否开通，以及您的账号是否通过了实名认证同时账户余额大于100元。

部署成功后，系统会返回集群ID，如下图所示。

```
...
Outputs:
cluster_id = [
  c0f2e04c77e234*****
]
...
vswitch_ids = [
  vsw-bp1c3hfc618*****
]
```

Kubernetes的*Kube Config*文件会存储在`~/.kube`目录下。您可以登录[容器服务控制台](#)查看通过Terraform创建的Kubernetes集群。

您可以通过以下参数自定义您的Kubernetes集群：

- `worker_instance_type`: Worker实例规格
- `worker_disk_category`: Worker实例系统盘
- `worker_disk_size`: Worker实例系统盘容量
- `ecs_password`: Worker实例登录密码
- `k8s_worker_number`: Worker实例数量
- `k8s_name_prefix`: 集群名称前缀

4. 执行以下命令销毁创建的集群。

```
cd ~/terraform-kubernetes-wordpress/kubernetes
terraform destroy
```

1.3 使用kubectl管理Kubernetes集群

kubectl是Kubernetes官方提供的管理k8s集群的命令行工具。您可以通过使用kubectl在Kubernetes集群上部署应用，并和容器完成交互。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（`tutorial-kubectl-deploy-app`）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过kubectl进行集群管理、应用部署等一系列操作。

在本教程中，您将完成以下操作：

- 使用kubect部署应用
- 使用kubect和容器交互

前提条件

在开始之前，确保您的账号下已创建一个Kubernetes集群。

执行以下命令，查看您的账号下是否已经有Kubernetes集群：

```
aliyun cs GET /clusters --output cols=cluster_id,region_id,cluster_type,state,name rows=[]
```



说明：

您可以登录[容器服务控制台](#)，创建Kubernetes集群。

部署应用

执行以下命令，直接向您的Kubernetes集群发起一个应用部署：

```
kubectl run nginx --image=registry.cn-shanghai.aliyuncs.com/jovi/nginx:alpine
```

如下输出，则说明部署成功。

```
deployment.apps/nginx created
```



说明：

- 本教程中部署nginx。
- 执行kubectl get deploy命令查看应用部署情况。

使用kubectl和容器交互

完成以下操作，实现nginx和容器的交互。

1. 执行以下命令，获取Pod名称：

```
export POD_NAME=$(kubectl get pods -o go-template --template '{{range .items}}{{.metadata.name}}{\n}}')
```

```
echo Name of the Pod: $POD_NAME
```

2. 执行以下命令，进入容器：

```
kubectl exec -it $POD_NAME sh
```

3. 执行以下命令，测试nginx的部署情况：

```
wget 127.0.0.1
```



说明：

如下输出，说明nginx已经成功运行。

```
Connecting to 127.0.0.1 (127.0.0.1:80)
index.html          100% |
*****
612    0:00:00 ETA
```

4. 执行以下命令，退出容器：

```
exit
```

1.4 在Alibaba Cloud Shell中快速体验Fun/fcli工具

您可以在Alibaba Cloud Shell中，快速体验用于函数计算开发的Fun/fcli工具。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-fc-cli）中集成。您可以通过点击代码块右上角的试用按钮，快速理解并体验Fun/fcli两种工具在函数计算平台完成从应用的创建、函数的增删改查全生命周期管理，以及后续关联云产品的配置等操作。

在本教程中涉及的两用于函数计算开发的工具的介绍：

- Fun介绍：Fun是一个用于支持Serverless应用部署的工具，能帮助您便捷地管理函数计算、API 网关、日志服务等资源。它通过一个资源配置文件（template.yml），协助您进行开发、构建、部署操作。
- fcli介绍：fcli是阿里云函数计算的命令行工具，帮助您便捷的管理函数计算中的资源。本教程带领您通过fcli命令完成函数计算的创建，修改，应用的部署等相关工作。

使用Fun工具部署应用

在Alibaba Cloud Shell中，为您内置了授权，无需安装和进行额外配置，您可以直接使用Fun工具来进行函数计算的相关操作。您可以直接执行fun --help来获取所有可执行命令的详细信息。

本教程使用的demo示例，已经加载到Alibaba Cloud Shell的教程目录中。您可以执行ls查看当前目录下所有文件。

执行以下命令，部署本地函数到函数计算。

```
fun deploy
```

初次使用fcli

在Alibaba Cloud Shell中，为您内置了授权，无需安装和进行额外配置，您可以直接使用fcli工具来进行函数计算的相关操作。您可以直接执行fcli --help来获取fcli所有可执行命令的详细信息。

命令如下：

```
alias          alias related operation
config        Configure the fcli
function      function related operation
help          Help about any command
service       service related operation
shell        interactive shell
sls           sls related operation
trigger       trigger related operation
version       fcli version information
```

Service相关命令

您可以通过执行fcli service --help命令来查看fcli提供了哪些service命令。



说明：

本教程中，类似myService命名的示例信息替换为您的真实信息。

· 创建服务：

1. 新建一个不带高级配置的服务：

```
fcli service create -s myService
```

2. 新建一个带高级配置的服务：

```
fcli service create -s myService -d 'my description' -p my-log-project -l my-log-store -r acs:ram::myAccountId:role/myRoleName
```

您可以通过以下参数自定义您的配置：

- myService：服务的名称。
- my description：指定服务对应的描述信息。
- my-log-project：指定服务对应的日志项目。
- my-log-store：指定服务对应的日志存储。
- myAccountId：账号ID。
- myRoleName：指定服务对应的角色。

- 列出当前环境变量配置的region下的所有服务。

```
fcli service list
```

- 获取单个服务的详细信息。

```
fcli service get -s myService
```

- 删除单个服务。

```
fcli service delete -s myService
```

- 获取更新服务时的所有可修改参数。

```
fcli service update --help
```

Function 相关命令

您可以通过 `fcli function --help` 来查看 `fcli` 提供了哪些 `function` 命令。

- 获取某个服务下的函数列表：

```
fcli function list -s myService
```

- 创建函数：

```
// 在相应service目录下
>>> fcli f create -s myTestService201 -f helloworld -t nodejs8 -h
helloworld.handler -b clitest20190109 -o helloworld.js //代码存储在oss
上, -t指定runtime, -h指定函数入口, -b指定代码所在的oss bucket, -o 指定了代码在
bucket中的object key
>>> fcli f create -s myTestService201 -f hello -t nodejs6 -h hello.
handler -d src -m 512 //代码存储在本地, -d指定了代码所在目录, -m设置函数执行的
内存大小
>>> fcli f create -s myTestService201 -f code.zip -t nodejs6 -h
hello.handler -d src -m 512 nodejs6 //代码在本地的code.zip中
```

- 更新函数：

```
// 在相应service目录下
>>> fcli f update -s myTestService201 -f helloworld -t nodejs8 -h
helloworld.handler -b clitest20190109 -o helloworld.js //代码存储在
oss上, -t指定runtime, -b指定代码所在的oss bucket, -o 指定了代码在bucket中的
object key
>>> fcli f update -s myTestService201 -f hello -t nodejs6 -h hello.
handler -d src -m 512 //代码存储在本地, -d指定了代码所在目录, -m设置函数执行的
内存大小
>>> fcli f update -s myTestService201 -f code.zip -t nodejs6 -h
hello.handler -d src -m 512 //代码在本地的code.zip中
```

- 触发函数

```
fcli f invoke -s myTestService201 -f hello
```



说明：

```
您可以通过 fcli f invoke -s myTestService201 --help 来查看 fcli 提供了哪些 invoke function 命令。
```

Trigger 相关命令

您可以通过执行 `fcli trigger` 命令来查看 `fcli` 提供了哪些 `trigger` 命令。

log 相关命令

执行以下命令查看 SLS 下的日志项目：

```
fcli sls project list
```

相关文档

- 关于 `fcli` 使用的更多信息，请参见[帮助文档中心](#)。
- 关于 Fun 使用的更多信息，请参见[帮助文档中心](#)。

2 云资源管控

2.1 使用Alibaba Cloud CLI管理弹性容器ECI实例

您可以通过使用Alibaba Cloud CLI调用阿里云ECI API，管理您的弹性容器实例（Elastic Container Instances, ECI）。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-eci-cli）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过Alibaba Cloud CLI管理您的ECI实例，包括：创建ECI容器组、获取ECI容器组信息、删除ECI容器组等操作。

前提条件

在使用Alibaba Cloud CLI调用ECI API之前，您需要确保：

- 已经开通ECI服务，并拥有访问您相应云资源的权限。



说明：

弹性容器实例(ECI)默认使用AliyunECIContainerGroupRole角色来访问您在其他云产品中的资源。

- 在当前ECI可用Region拥有VPC网络，且在ECI可用区内有VSwitch。
- 在当前ECI可用Region拥有安全组。

相关API

您可以通过使用Alibaba Cloud CLI调用以下API，对ECI实例进行管理。

| API名称 | 说明 |
|---|------------|
| CreateContainerGroup | 创建一个容器组。 |
| DescribeContainerGroups | 批量获取容器组信息。 |
| DescribeContainerLog | 获取容器组日志信息。 |
| DeleteContainerGroup | 删除一个容器组。 |
| ExecContainerCommand | 在容器内部执行命令。 |



说明：

执行aliyun eci help命令，获取通过阿里云命令行工具CLI管理弹性容器实例的可用方法。

典型案例

在创建ECI实例之前，您需要先获取VSwitchId和SecurityGroupId。

1. 查询已创建的交换机的列表。

```
aliyun vpc DescribeVSwitches | jq .VSwitches.VSwitch | jq .[] | jq .ZoneId
```



说明:

若上述命令输出为空，表示该Region不存在VPC网络。您需要登录[VPC控制台](#)创建相关VPC。若交换机都不在[ECI地域和可用区](#)内，您还需要[创建VSwitch](#)。

2. 假设第一个交换机符合要求，执行以下命令，获取VSwitchId。

```
VSwitchId=`aliyun vpc DescribeVSwitches | jq .VSwitches.VSwitch | jq .[0] | jq -r .VSwitchId`
```



说明:

选择当前Region下，ECI可用区范围内的交换机。

3. 查询已创建的安全组的基本信息。

```
aliyun ecs DescribeSecurityGroups | jq .SecurityGroups.SecurityGroup | jq .[] | jq .SecurityGroupId
```



说明:

若上述命令输出为空，您需要登录[云服务器控制台](#)创建安全组。

4. 执行以下命令，获取SecurityGroupId。

```
SecurityGroupId=`aliyun ecs DescribeSecurityGroups | jq .SecurityGroups.SecurityGroup | jq .[0] | jq -r .SecurityGroupId`
```

5. 使用Alibaba Cloud CLI管理弹性容器ECI实例。

示例如下:

- 创建容器组。

```
aliyun eci CreateContainerGroup --SecurityGroupId ${SecurityGroupId} --VSwitchId ${VSwitchId} --ContainerGroupName test-eci --
```

```
Container.1.Image nginx --Container.1.Name nginx --Container.1.Cpu
1 --Container.1.Memory 2
```

类似如下输出：

```
{"ContainerGroupId":"eci-bp19vpbluci6jm3wblj2","RequestId":"
F1F69DA4-4952-441D-A164-7C8E3EDF0DC0"}
```

- 批量获取容器组信息。

```
aliyun eci DescribeContainerGroups --ContainerGroupName test-eci
| jq
```

- 获取容器组日志信息。

```
ContainerGroupId=`aliyun eci DescribeContainerGroups --ContainerG
roupName test-eci | jq .ContainerGroups | jq .[] | jq -r .
ContainerGroupId`
aliyun eci DescribeContainerLog --ContainerGroupId ${ContainerG
roupId} --ContainerName nginx
```

- 删除容器组。

```
aliyun eci DeleteContainerGroup --ContainerGroupId ${ContainerG
roupId}
```

2.2 使用Packer创建ECS自定义镜像

Packer是一款轻量级的镜像定义工具，能够运行在常用的主流操作系统上。本教程介绍了通过创建一个JSON格式的Packer模板文件来构建ECS自定义镜像。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-hashicorp-packer）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过Packer模板文件来构建您的ECS自定义镜像。

安装Packer

您可以在 [Packer官网](#) 页面获取不同平台最新版本的Packer，选择下载与您操作系统对应的版本。并通过以下示例或者访问[Packer官方安装说明](#)来安装Packer。

```
mkdir -p $HOME/bin
curl https://shell-image-softwares-cn-shanghai.oss-cn-shanghai.
aliyuncs.com/packer/packer_1_3_5.tar.gz | tar zxv -C $HOME/bin
```

定义Packer模板

使用Packer创建自定义镜像时，需要先创建一个JSON格式的模板文件。在该模板文件中，您需要指定创建自定义镜像的[Alicloud Image Builder（生成器）](#)和[Provisioners（配置器）](#)。Packer具有多种配置器，可用于配置自定义镜像的内容生成方式。

本教程以常用的Shell配置器为例，定义了一个名为alicloud.json的Packer模板，模板内容如下：

```
{
  "variables": {
    "access_key": "{{env `ALICLOUD_ACCESS_KEY`}}",
    "secret_key": "{{env `ALICLOUD_SECRET_KEY`}}"
  },
  "builders": [{
    "type": "alicloud-ecs",
    "access_key": "{{user `access_key`}}",
    "secret_key": "{{user `secret_key`}}",
    "region": "cn-beijing",
    "image_name": "packer_basic",
    "source_image": "centos_7_02_64_20G_alibase_20170818.vhd",
    "ssh_username": "root",
    "instance_type": "ecs.n1.tiny",
    "internet_charge_type": "PayByTraffic",
    "io_optimized": "true"
  }],
  "provisioners": [{
    "type": "shell",
    "inline": [
      "yum install redis.x86_64 -y"
    ]
  }
]}
}
```



说明：

- 您可以执行 `cat alicloud.json` 命令查看模板详情。
- 本示例中，将创建一个含Redis的自定义镜像。

您可以通过以下参数自定义您的模板文件：

| 参数名 | 说明 |
|----------------------|-------------------------|
| access_key | 您的AccessKey ID。 |
| secret_key | 您的AccessKey Secret。 |
| region | 创建自定义镜像时使用临时资源的地域。 |
| image_name | 自定义镜像的名称。 |
| source_image | 基础镜像的名称，可以从阿里云公共镜像列表获得。 |
| instance_type | 创建自定义镜像时生成的临时实例的类型。 |
| internet_charge_type | 创建自定义镜像时临时实例的公网带宽付费类型。 |
| provisioners | 创建自定义镜像时使用的Packer配置器类型。 |

使用Packer创建自定义镜像

在使用上述Packer模板文件创建ECS自定义镜像之前，您需要确保您已有ECS的相关操作权限。如果已开放权限，在Alibaba Cloud Shell环境下，可以直接执行如下命令来构建镜像。

```
packer build alicloud.json
```

示例运行结果如下：

```
alicloud-ecs output will be in this color.
==> alicloud-ecs: Prevalidating alicloud image name...
alicloud-ecs: Found image ID: centos_7_02_64_20G_alibase_20170818.vhd
==> alicloud-ecs: Start creating temporary keypair: packer_59e44f40-
c8d6-0ee3-7fd8-b1ba08ea94b8
==> alicloud-ecs: Start creating alicloud vpc
-----
==> alicloud-ecs: Provisioning with shell script: /var/folders/3q/
w38xx_js6cl6k5mwkrqsnw7w0000gn/T/packer-shell257466182
alicloud-ecs: Loaded plugins: fastestmirror
-----
alicloud-ecs: Total                                     1.3
MB/s | 650 kB 00:00
alicloud-ecs: Running transaction check
-----
==> alicloud-ecs: Deleting temporary keypair...
Build 'alicloud-ecs' finished.
==> Builds finished. The artifacts of successful builds are:
--> alicloud-ecs: Alicloud images were created:
cn-beijing: m-2ze12578be1oa4ovs6r9
```

更多信息

- 获取更多信息，请参见阿里云GitHub Packer仓库[packer-provider](#)。
- 了解更多Packer使用详情，请参见[Packer官方文档](#)。

2.3 复制ECS安全组

云命令Cloud Shell中预装了一个复制ECS安全组的脚本。您可以通过运行该脚本快速创建一个新的安全组。

背景信息

安全组是一种虚拟防火墙，具备状态检测和数据包过滤功能，用于在云端划分安全域。您可以通过配置安全组规则，允许或禁止安全组内的ECS实例对公网或私网的访问。

操作步骤

1. 在浏览器中输入<https://shell.aliyun.com>或在[OpenAPI Explorer](#)中进入Cloud Shell页面。

2. 执行以下命令，复制脚本。

```
git clone https://code.aliyun.com/labs/tutorial-ecs-copy-security-group.git
```

3. 执行以下命令进入 `tutorial-ecs-copy-security-group` 目录。

```
cd tutorial-ecs-copy-security-group
```

4. 执行以下命令复制安全组。

```
sh CloneSecurityGroup.sh
```

5. 根据提示完成以下操作，创建一个和源安全组规相同的安全组：

- 输入源安全组的所属地域，例如 `cn-hangzhou`。您可以直接按 `Enter` 键查看地域信息。
- 输入源安全组的ID。

您可以直接按 `Enter` 键查看指定地域下所有的安全组。

```
1. Please enter RegionID where the source SecurityGroup is located, or press "Enter" for more Region information.
Please type RegionID to continue: cn-hangzhou
2. Please enter source SecurityGroupId or press "Enter" to get SecurityGroupId.

Please type SecurityGroupId to continue:
SecurityGroupId | SecurityGroupName | Description
-----|-----|-----
sg-bpla3uv8r1 | | 
sg-bp1190c50 | secu | 2
sg-bp11j9xpp | secu | 2
sg-bp168k8dw | sg-b | 3 | System created security group.
sg-bp16tmhcb | sg-b | 1 | System created security group.
sg-bp17cfy7s | sg-b | 9 | System created security group.
sg-bp13pwx19 | VPC | 
sg-bp11um4k0 | wugo | 
sg-bp1ddqy41 | wugo | 
sg-bp1hbr8zp | xian | test

Please type SecurityGroupId to continue: sg-bpla3uv8r1ryyq3e9bsfh
```

- 输入目标新创建的安全组的所属地域，例如 `cn-shanghai`。您可以直接按 `Enter` 键查看地域信息。
- 输入目标新创建的安全组的所属VPC ID。您可以直接按 `Enter` 键查看目标地域下的VPC信息。

```

3. Please type the target RegionID or press "Enter" to clone SecurityGroup in the same region.

Please type RegionID to continue: cn-shanghai
4. Please type the target VpcId or press "Enter" to get VpcId.

Please type VpcId to continue:
VpcId          | VpcName          | Description
-----
vpc-uf614ufd   | doctest-1       |
vpc-uf6mb30e   | miniapp_vpc_test |
vpc-uf604yqg   | miniapp_vpc_prod |
vpc-uf6o8d1d   | yyy             |
vpc-uf6p2ul9   | www             |
vpc-uf6qpogy   |                 |
vpc-uf62xei9   | CL-ggg         |
vpc-uf6nr5j1   | cl-HaVip       |
vpc-uf6oxj71   |                 |
vpc-uf66v509   | drntest        |

Please type VpcId to continue: vpc-uf614ufdj13n8

```

e) 输入要创建的安全组名称和描述。您可以按Enter键略过名称和描述，直接创建安全组。

```

5. Please type the target SecurityGroupName or press "Enter" to skip.

Please type SecurityGroupName to continue: ingressTraffic
6. Please type the target Description or press "Enter" to skip.

Please type Description to continue: copyfromsg-bpla3uv8rrypq3e9bsfh

Cloning the specified security group, please wait a moment -

Congratulations! You successfully cloned a SecurityGroup: sg-uf61a39n0ss2xt3eczxo

```

预期结果

安全组创建成功后，系统会返回安全组ID。您可以执行以下命令查看创建的安全组规则详情。

```
aliyun ecs DescribeSecurityGroupAttribute --RegionId cn-shanghai --
SecurityGroupId sg-adf13*****
```

其中：

- RegionId是已创建的安全组的所属地域。
- SecurityGroupId是已创建的安全组ID。

3 解决方案

3.1 基于Terraform部署Kubernetes托管版集群及Wordpress应用

您可以通过Terraform部署一个阿里云容器服务的Kubernetes托管版集群，并在该集群上部署一个Wordpress 应用。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（[terraform-kubernetes-wordpress](#)）中集成，通过点击代码块右上角试用按钮，快速体验通过Terraform来自动创建、编排和管理容器服务，以及完成在容器集群上应用的自动部署。

完成本教程后，您会创建以下资源。其中，容器服务没有任何附加费用，您只需要支付所使用资源（云服务器、负载均衡等）的费用。

- Worker 实例（ECS）
 - 实例规格：ecs.n2.medium
 - 实例数量：3
 - 系统盘：20G 高效云盘
- 负载均衡
 - 实例数量：3
 - 付费模式：按量付费
- 弹性公网IP
 - 实例数量：1
 - 付费模式：使用流量计费
- NAT网关
 - 实例数量：1
 - 付费模式：按量付费

具体计费信息，参见 [ECS计费概述](#)、[负载均衡按量计费](#)、[弹性公网IP按量计费](#)、[NAT网关按量计费](#)。

使用限制

在开始使用本教程之前，确保您已经了解以下限制并满足相关要求：

- 保证您的账户有100元的余额并通过实名认证，否则无法创建按量付费的ECS实例和负载均衡。

- 随集群一同创建的负载均衡实例只支持按量付费的方式。
- Kubernetes集群仅支持专有网络VPC。
- 您的每个账号默认可以创建的云资源有一定的配额，如果超过配额，集群创建失败。如果您需要提高配额，请提交工单申请。
 - 每个账号默认最多可以创建100个安全组。
 - 每个账号默认最多可以创建60个按量付费的负载均衡实例。
 - 每个账号默认最多可以创建20个EIP。
- 在开始之前，确保您已开通了以下云服务：
 - [容器服务](#)
 - [资源编排（ROS）服务](#)
 - [弹性伸缩（ESS）服务](#)
 - [访问控制（RAM）服务](#)
 - [弹性公网IP服务](#)
 - [NAT网关服务](#)
 - [负载均衡服务](#)

创建托管版Kubernetes集群

若您已有Kubernetes集群，需要配置集群凭证。您可以登录[容器服务控制台](#)，将您的集群凭证复制到`~/.kube/config`文件中。

完成以下操作，创建Kubernetes集群：

1. 执行以下命令定位到用来创建Kubernetes集群的Terraform模板的目录。

```
cd ~/terraform-kubernetes-wordpress/kubernetes
```

2. 执行init命令加载[Alibaba Cloud Providers](#)。

```
terraform init
```

3. 执行以下命令部署集群。

```
terraform apply
```



说明：

如果出现 `ErrManagedKubernetesRoleNotAttach` 的错误，请检查所需服务是否开通，以及您的账号是否通过了实名认证同时账户余额大于100元。

部署成功后，系统会返回集群ID，控制台显示如下输出。

```
...
Outputs:
cluster_id = [
  c0f2e04c77e234*****
]
...
vswitch_ids = [
  vsw-bp1c3hfcd6l8*****
]
```

Kubernetes的*Kube Config*文件会存储在`~/.kube`目录下。您可以登录[容器服务控制台](#)查看通过Terraform创建的Kubernetes集群。

您可以通过以下参数自定义您的Kubernetes集群：

- `worker_instance_type`: Worker实例规格
- `worker_disk_category`: Worker实例系统盘
- `worker_disk_size`: Worker实例系统盘容量
- `ecs_password`: Worker实例登录密码
- `k8s_worker_number`: Worker实例数量
- `k8s_name_prefix`: 集群名称前缀

部署Wordpress应用

完成以下操作，将Wordpress应用部署到之前创建的Kubernetes托管版集群中。

1. 执行以下命令定位到用来部署Wordpress应用的Terraform模板的目录。

```
cd ~/terraform-kubernetes-wordpress/wordpress
```

2. 执行init命令加载kubernetes providers。

```
terraform init
```

3. 执行以下命令部署应用。其中，需要显式指定MySQL的密码，也可以指定需要的Wordpress和MySQL的版本。

```
terraform apply -var 'mysql_password=Test12345'
```

```
terraform apply -var 'mysql_version=5.6' -var 'wordpress_version=4.7.3' -var 'mysql_password=Test12345'
```



说明：

如果出现 `ErrManagedKubernetesRoleNotAttach` 的错误，请检查所需服务是否开通，以及您的账号是否通过了实名认证同时账户余额大于100元。

部署成功后，控制台显示如下输出。

```
...
Outputs:
slb_ip = 35.197.xx.xx
```

其中，`slb_ip`表示应用负载均衡的公网IP。在浏览器中访问该IP地址，可以看到您部署的Wordpress应用。

销毁应用

按照以下步骤完成操作，销毁您部署的应用。

1. 执行以下命令定位到用来部署Wordpress应用的Terraform模板的目录。

```
cd ~/terraform-kubernetes-wordpress/wordpress
```

2. 执行以下命令销毁您部署的Wordpress应用。

```
terraform destroy -var 'mysql_password=Test12345'
```

3. 执行以下命令定位到用来创建Kubernetes集群的Terraform模板的目录。

```
cd ~/terraform-kubernetes-wordpress/kubernetes
```

4. 执行以下命令销毁您创建的Kubernetes集群。

```
terraform destroy
```

3.2 基于弹性容器实例部署Parse Server应用

您可以通过使用Alibaba Cloud CLI创建ECI容器组，同时在该容器组上部署Parse Server应用，并提供公网访问。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（`tutorial-eci-parse-server`）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过Alibaba Cloud CLI部署Parse Server应用。

在本教程中，您将完成以下操作：

1. 网络配置
2. 部署应用
3. 销毁应用

完成本教程后，您会创建以下资源：

- ECI容器组
 - 容器数量：3
 - 容器规格：1vCPU、1G内存
- 弹性公网IP
 - 实例数量：1
 - 付费模式：使用流量计费

具体计费信息，请参见：[ECI计费概述](#)、[弹性公网IP按量计费](#)。

前提条件

在开始使用本教程之前，确保您已经了解以下限制并满足相关要求：

- 确保您已开通了以下云服务：
 - [弹性容器实例（ECI）服务](#)
 - [访问控制（RAM）服务](#)
 - [弹性公网 IP 服务](#)
- 请保证您的账户有大于100元的余额并通过实名认证，否则无法创建ECI实例。

网络配置

在部署Parse Server应用之前，您首先需要配置网络环境。

1. 执行以下命令定位到用来配置网络环境的Terraform模板的目录。

```
cd ~/tutorial-eci-parse-server/network
```

2. 执行init命令加载[Alibaba Cloud Providers](#)。

```
terraform init
```

3. 执行以下命令配置网络环境。

```
terraform apply
```

部署成功后，您可以看到控制台显示如下输出：

```
...
Outputs:
eip_ip = 106.14.2.**
eip_id = eip-uf6n81mgxtwl*****
region_id = cn-hangzhou
securitygroup_id = sg-uf631pc9pwu*****
vpc_id = vpc-uf6rrgilzrug3znr*****
vswitch_id = vsw-uf6jxv84mbf427h*****
```

```
zone_id = cn-hangzhou-b
```

您可以通过指定以下的参数来自定义您的网络配置：

region 地域，默认值为cn-hangzhou。

availability_zone可用区，默认值为cn-hangzhou-b。

vpc_cidr VPC 网段，默认值为172.16.0.0/12。

vswitch_cidr 交换机网段，默认值为172.16.0.0/21。

4. 执行以下命令，将新开通的网络参数输出到一个临时文件中，后续教程会使用到这个配置文件。

```
terraform output > net_conf
```

部署Parse Server应用

本教程中提供了`eci-create.sh`脚本，它会读取之前生成的网络配置文件，并创建ECI容器组来部署Parse Server应用。关于ECI的CLI操作，请参见使用Alibaba Cloud CLI管理弹性容器实例ECI。

执行以下命令，定位到教程目录（`tutorial-eci-parse-server`）中，并部署ParseServer应用。

```
cd ~/tutorial-eci-parse-server
bash eci-create.sh network/net_conf
```

其中，

- `network/net_conf` 指向上个步骤输出的保存网络参数的配置文件。
- 创建的容器组包含三个容器实例，一个用来运行MongoDB，一个用来运行Parse Server，最后一个用来运行Parse Dashboard。

部署成功后，您可以看到控制台显示如下输出：

```
{
  "ContainerGroupId": "eci-uf6fonngghi50val****",
  "RequestId": "3061989B-C3D2-41BD-B30F-*****"
}
```

此时，您可以登录ECI控制台查看您创建的容器组。



说明：

确认创建成功后，您可以通过`http://{eip的ip address}:4040`访问Parse DashBoard。

销毁应用

您可以通过删除容器组和网络配置来销毁应用。

1. 执行以下命令删除容器组。

```
aliyun eci DeleteContainerGroup \  
  --RegionId cn-hangzhou \  
  --ContainerGroupId eci-uf6fonngghi50val****
```

其中，ContainerGroupId为您上一步创建的容器组ID。

2. 执行以下命令定位到用来配置网络环境的Terraform模板的目录。

```
cd ~/tutorial-eci-parse-server/network
```

3. 执行以下命令删除网络配置。

```
terraform destroy
```

3.3 通过Helm部署Jenkins

Helm项目提供了一个统一的软件打包方式，支持版本控制，可以大大简化Kubernetes应用分发与部署中的复杂性。本教程介绍了如何在Cloud Shell中通过Helm部署Jenkins。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-helm-jenkins）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过Helm部署Jenkins。

在本教程中，您将完成以下操作：

- [配置Helm](#)
- [部署Jenkins](#)
- [测试验证](#)
- [删除应用](#)

前提条件

在开始前，确保您已满足以下条件：

- 已经部署了一个Kubernetes集群，并且版本为1.8.4或更高。



说明：

如果您的集群版本不满足要求，请进行升级。

- 将Kubernetes集群的配置文件存放在Cloud Shell的 `~/.kube/config` 目录中。

配置Helm

执行以下命令使用阿里云镜像来配置Helm。

```
helm init --upgrade -i registry.cn-hangzhou.aliyuncs.com/google_containers/tiller:v2.12.1 --stable-repo-url https://kubernetes.oss-cn-hangzhou.aliyuncs.com/charts
```

安装成功后，您可以使用以下命令管理Helm：

- 执行以下命令，查看在存储库中可用的所有Helm charts：

```
helm search
```

- 执行以下命令，更新charts列表：

```
helm repo update
```

- 执行以下命令，查看在集群上安装的charts列表：

```
helm ls
```

部署Jenkins

执行以下命令部署Jenkins：

```
helm install --name my-jenkins --set Persistence.Enabled=false stable/jenkins
```



说明：

Persistence.Enabled设置为false，会禁用PersistentVolume。在您的生产环境下，可以创建一个NAS作为PersistentVolume。

测试验证

完成以下操作，测试部署的Jenkins：

1. 执行以下命令，获取账号和登录密码：

```
helm install --name my-jenkins --set Persistence.Enabled=false stable/jenkins
```

2. 执行以下命令，查看登录地址：

```
export SERVICE_IP=$(kubectl get svc --namespace default my-jenkins --template '{{ range (index .status.loadBalancer.ingress 0) }} {{ . }}{{ end }}') && echo http://$SERVICE_IP:8080/login
```



说明：

访问地址可能并不能马上生效，需要等待负载均衡等资源启动完成。您可以使用如下命令，来检查公网IP是否已经被分配。

```
kubectl get svc --namespace default my-jenkins
```

3. 访问输出在Terminal中的地址，即可看到您刚刚部署完成的Jenkins，可使用获取的账号和密码登录。

删除应用

执行以下命令，删除部署的Jenkins：

```
helm delete --purge my-jenkins
```

3.4 快速部署智能相册解决方案应用

本教程介绍了通过Alibaba Cloud Shell快速部署智能相册解决方案应用，基于内置的基础应用使您快速体验阿里云智能媒体管理（Intelligent Media Management，简称 IMM）的强大功能。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-smart-photos）中集成。您可以通过点击代码块右上角的试用按钮，快速体验通过Alibaba Cloud Shell部署智能相册解决方案应用。

完成本教程后，您会收获：

1. 获取到智能相册解决方案应用的URL。

2. 快速体验IMM的强大功能：

- 上传图片到OSS。
- 智能排序列表。
- 智能分类列表。

- 人脸分组：

- [CreateFaceSet](#)：创建人脸分组数据集。
- [IndexFace](#)：检测相片的人脸，并把人脸信息加入到数据集。
- [GroupFaces](#)：对元数据集中的人脸进行分组聚类操作。
- [GetFaceSetDetail](#)：获取元数据集中的人脸分组详细信息。

- 地点分组：

- [OSS API](#)
- [高德](#)

解决方案

随着存储设备成本的降低，很多用户存储了大量的照片，因此也带来了巨大的管理成本。IMM提供了图像识别、自动打标、智能分类展示等功能，帮助用户大幅降低照片的管理成本。

智能相册解决方案，内置相册的基础应用，整体架构采用Serverless方案，包含函数计算、表格存储、对象存储OSS、IMM等阿里云产品。

前提条件

在部署智能相册解决方案应用之前，您需要开通以下服务：

- [函数计算 \(FC\)](#)
- [表格存储 \(OTS\) 服务](#)
- [对象存储 \(OSS\) 服务](#)
- [智能媒体 \(IMM\) 服务](#)
- [访问控制 \(RAM\) 服务](#)



说明：

因为涉及到地理位置服务，您需要申请高德[的APPKEY](#)。

部署应用

完成以下操作，获取智能相册解决方案应用的URL。

1. 执行以下命令，克隆智能相册代码。

```
cd ~ && git clone https://code.aliyun.com/solutions/smart-photos.git
&& cd smart-photos
```

2. 执行以下命令，配置脚本。

```
./configure.sh
```



说明：

- region、Account ID配置默认值。
- 默认配置MAP_APPKEY的值为171008d2c0973387f6e1b43729db107a。

3. 执行以下命令，通过fun工具部署应用代码到函数计算。

```
fun deploy
```

4. 执行以下命令，访问返回的应用地址。

```
./qrcode.sh
```

通过点击链接或者扫描二维码的方式，进入智能相册应用，体验IMM的强大功能。

3.5 快速体验趣视频解决方案

本教程为阿里云趣视频APP后台服务接口集成教程，主要用于解决趣视频App与视频点播服务之间数据交互、业务逻辑处理，提供运营对数据进行管理。包括随机生成用户，获取视频列表，获取凭证信息，上传视频，各种回调信息处理等能力。

教程介绍

以下步骤及示例均已在Alibaba Cloud Shell教程目录（tutorial-aliyun-quvideo-appserver）中集成。您可以通过点击代码块右上角的试用按钮，快速体验趣视频解决方案。

在本教程中提到的趣视频解决方案是完整的短视频端到端的解决方案，涵盖短视频整个生命周期：视频生产（采编）、媒资管理（存储、转码、审核）、分发消费（CDN加速、播放）等。同时提供产品级的UI和交互、业务层源码，方便企业快速搭建短视频APP。解决方案提供丰富的视频功能以及产品级的源码、交互和样式。客户端提供了强大的视频采集和编辑能力：实时音乐、变速、美颜贴纸特效、字幕、MV等能力。同时在编解码、视频渲染上性能优异。后台服务端提供了丰富的API和管理控制台的能力，根据自身需求灵活调用，降低开发成本，方便快速集成和上线。主要涉及的技术有Spring Boot微服务框架、RDS (Mysql) 数据库、Mybatis持久层框架。

前提条件

在创建云资源之前，确保您已经开通了以下云服务：

- [云服务器（ECS）服务](#)

推荐配置：

- 计费方式：按量付费
- 地域：华东2（上海）
- CPU：2核
- 内存：4GB
- 实例类型：I/O优化
- 操作系统：Aliyun Linux 17.1 64位

- [云数据库 \(RDS MySQL版\) 服务](#)

推荐配置:

- 地域: 多可用区 (根据自身需求进行选择)
- 网络: 专有网络
- 数据库类型: MySQL 5.7 (最低5.6)
- 存储空间 :150GB (本地SSD盘)
- 规格 :2 核 4G
- 购买时长 :1年 (根据需求选择)

- [视频点播 \(VOD\) 服务](#)

- [对象存储 \(OSS\) 服务](#)

- [阿里云RAM账户配置](#)



说明:

- 以上服务开通过程中, 地区要保持一致。
- 确保您的账户余额大于100元。

创建云资源

完成以下操作, 快速创建ECS、RDS、VOD、RAM等云资源。

1. 执行以下命令, 定位到Alibaba Cloud Shell的教程目录 (tutorial-aliyun-quvideo-appserver) 中, 并安装云资源相关依赖。

```
cd ~/tutorial-aliyun-quvideo-appserver && pip install -r requirements.txt --user
```

2. 域名配置: 域名是视频点播服务的必填项。在应用部署的过程中, 会提示输入域名, 您需要先手动进行配置。更多信息, 请参见[添加域名](#)。

3. 执行以下命令, 下载趣视频的源码。

```
cloudshell-git-open https://code.aliyun.com/solutions/AlivcSolution_AppServer.git
```

4. 执行以下命令, 创建云资源。

```
python configuration/index.py
```

部署应用

完成以下操作, 部署应用到阿里云服务端。

1. 执行以下命令，定位到创建云资源时下载的源代码目录（AlivcSolution_AppServer）中，并对源代码进行构建。

```
cd ~/AlivcSolution_AppServer && mvn package
```

2. 执行以下命令，定位到教程目录（tutorial-aliyun-quvideo-appserver）中，并部署应用到阿里云服务端。

```
cd ~/tutorial-aliyun-quvideo-appserver && python deploy/index.py
```



说明:

示例项目中包含用户模块，其中，头像的url是为了快速部署应用，已经为您自动生成了一个url地址。同时，可支持自定义头像，通过修改在 `tutorial-aliyun-quvideo-appserver/template/application.properties.template` 目录中，`AVATARDOMAIN_NAME` 参数的地址。

部署成功后服务启动，会返回客户端访问地址，服务端接口访问地址，和管理后台访问地址。

服务访问

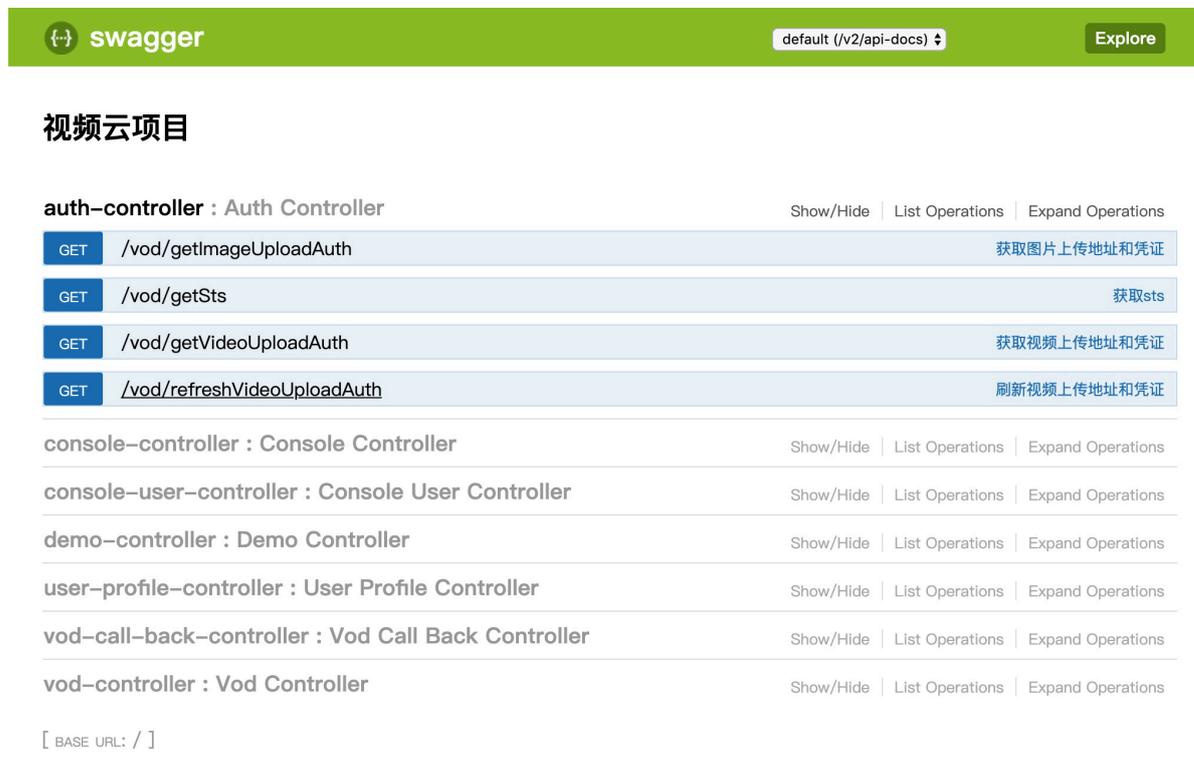
服务正常启动后，请根据以下规则进行服务访问测试。

· 服务端接口访问

根据应用部署成功之后返回的IP地址，即可根据如下规则测试趣视频服务的可用性。

```
http://{ip地址}:8080/swagger-ui.html
```

服务部署成功即可显示如下页面。



· 管理后台访问

根据应用部署成功之后返回的IP地址，即可根据如下规则测试趣视频管理后台的可用性。

```
http://{ip地址}:8080/resource/index.html#/
```

管理后台默认账户名：admin，密码：12345678。

- 客户端访问

- iOS

在`/AliyunVideoClient_Entrance/AlivcShortVideo/AlivcShortVideoFile/VideoSolution/Model`路径下，存在一个名称为`AlivcQuVideoServerManager.m`的文件。您需要将文件里的静态变量`kAlivcQuUrlString`替换为云服务器公网IP地址。

```
http://{ip地址}:8080
```

- Android

在`AlivcLittleVideo/src/main/java/com/aliyun/apsara/alivclittlevideo/constants`路径下，存在一个名称为`AlivcLittleServerApiConstants.class`的文件。您可以将文件里的`BASE_URL`替换为您的云服务器公网IP地址。

```
http://{ip地址}:8080
```



说明:

默认客户端使用Https访问，但是本次部署不涉及https，请直接使用http访问。

删除云资源

如您已经完成应用演示，执行以下命令删除已创建的云资源，避免不必要的计费。

```
cd ~/tutorial-aliyun-quvideo-appserver && python delete/index.py
```