

阿里云镜像服务使用

AUTHOR: 彭玲 TIME: 2021/11/26

阿里云镜像服务使用

背景描述

前置条件

制作镜像

1. 创建命名空间
2. 创建镜像仓库
3. 镜像构建

使用镜像

背景描述

在国内，由于 gfw 的原因，我们是访问不了 google 的镜像服务 (`k8s.gcr.io`) 的，因此很多时候我们会碰到类似这样的错误 `failed pulling image k8s.gcr.io/pause:3.1`。这个问题我们可以通过 阿里云容器镜像服务 来解决：利用阿里云的 海外机器 对 `k8s.gcr.io` 镜像服务进行镜像构建。

前置条件

1、阿里云账号

2、待制作镜像的 `Dockerfile`。例如，我的 GitHub 仓库 `google_containers` 的 `images/` 中提供了相关 `Dockerfile` 文件。

- `images/kube-apiserver/Dockerfile`

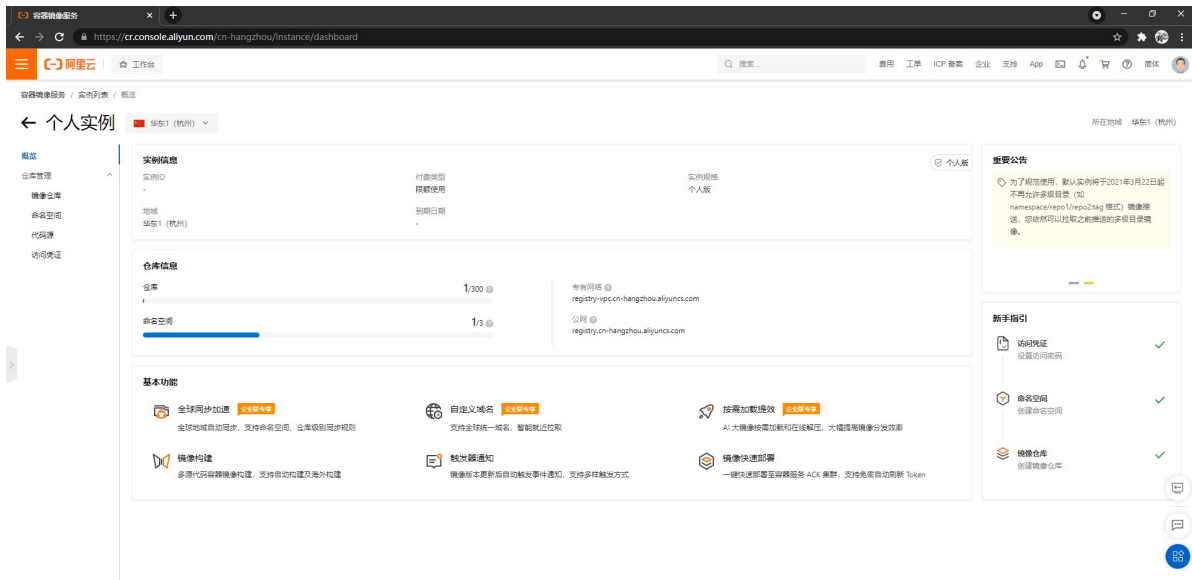
```
1 FROM k8s.gcr.io/kube-apiserver:v1.18.3
```

- `images/nginx-controller-alpha/Dockerfile`

```
1 FROM k8s.gcr.io/nginx-  
controller:v0.49.1@sha256:4080849490fd4f61416ad7bdba6fdd0ee58164f2e13df  
1128b407ce82d5f3986
```

制作镜像

阿里云 [容器镜像服务](#) - 实例列表 分为 个人版 和 企业版。其中，个人版 命名空间 3 个，仓库可以 300 个，够用了。



1. 创建命名空间

选择 **个人实例** - **仓库管理** - **命名空间** 子菜单，创建命名空间。



2. 创建镜像仓库

选择 **个人实例** - **仓库管理** - **镜像仓库** 子菜单，创建镜像仓库。

1 仓库信息

地域 华东1 (杭州)

* 命名空间 gcr_k8s_container

* 仓库名称 kube-apiserver 14/64
长度为2-64个字符, 可使用小写英文字母、数字, 可使用分隔符“_”、“-”、“.” (分隔符不能在首位或未位)

仓库类型 公开 私有

* 摘要 k8s.gcr.io/kube-apiserver 25/100
长度最长100个字符

描述信息
支持Markdown格式

2 代码源

下一步 取消

下一步, 选择代码源, 比如选择 GitHub 代码源 (首次使用需要授权)。例如:

- 选择 命名空间: julinpeng
- 选择 仓库: google_containers
- 构建设置: 勾选 海外机器构建

✓ 仓库信息

2 代码源

代码源 云Code GitHub Bitbucket 私有GitLab 本地仓库

julinpeng julinpeng google_containers

构建设置 代码变更自动构建镜像 海外机器构建 不使用缓存

构建规则设置 请在仓库创建完成后前往构建页面设置

上一步 创建镜像仓库 取消

3. 镜像构建

选择 镜像仓库 列表中的某个仓库，如 kube-apiserver 仓库名称。选择其下的 构建 子菜单，添加构建规则。

添加构建规则

* 类型: Tag

* Branch/Tag: master

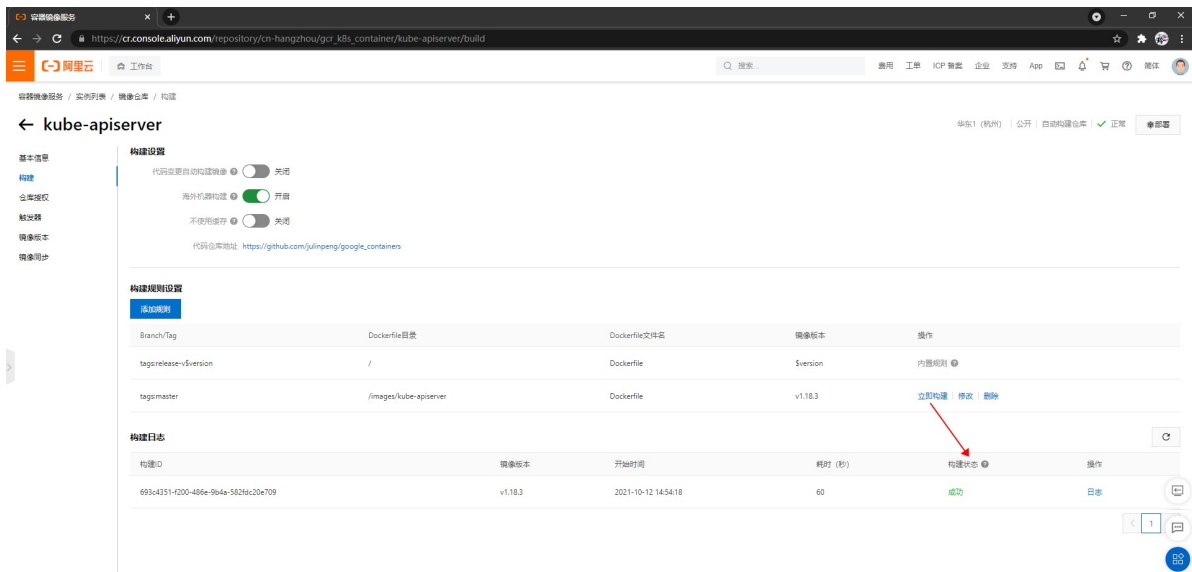
* Dockerfile目录: /images/kube-apiserver 22/128
1-128个字符，支持英文字母、数字、“-”、“_”、“.”、“/”

* Dockerfile文件名: Dockerfile 10/64
1-64个字符，支持英文字母、数字、“-”、“_”、“.”

* 镜像版本: v1.18.3 7/128
支持英文字母、数字、“-”、“_”、“.”

确定 取消

添加构建规则后，点击 立即构建，利用阿里云的海外服务器构建 google 的相关镜像。



使用镜像

镜像构建 成功 后，就可以通过阿里云 容器镜像服务 中的 镜像仓库 使用镜像了，使用的时候指定 镜像版本号 即可。

例如，使用 docker pull 拉取 kube-apiserver 镜像（版本号 v1.18.3）：

```
1 # docker pull registry.cn-hangzhou.aliyuncs.com/<命名空间>/<镜像名称>:<镜像版本号>
2 $ docker pull registry.cn-hangzhou.aliyuncs.com/gcr_k8s_container/kube-apiserver:v1.18.3
```

