

# 8.如何发布镜像

---

## 1.创建个人镜像仓库

创建命名空间

创建镜像仓库

管理镜像仓库

## 2.推送镜像

添加标签

登录镜像仓库

推送镜像

上一节我们学习了如何创建一个新的镜像，现在来学习如何发布这个镜像到镜像仓库。

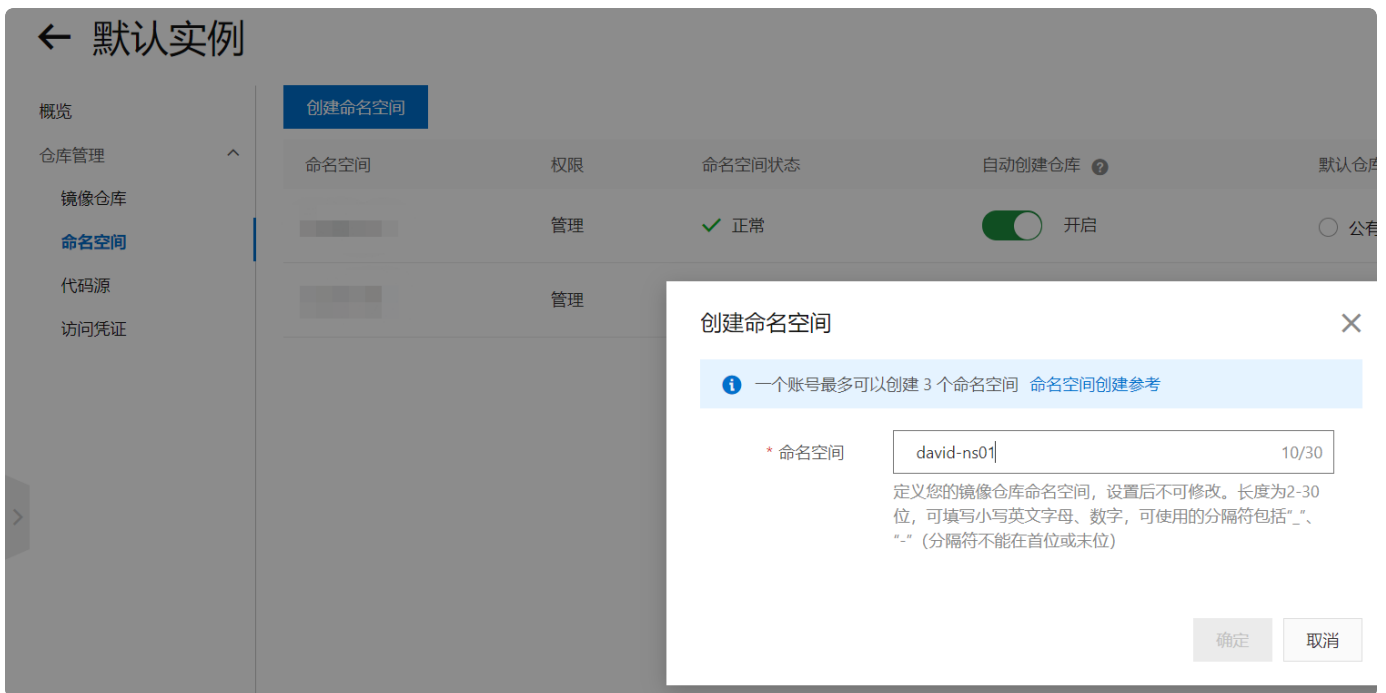
到目前为止，前面创建的镜像只存在于我们自己的电脑上。如果想要它可以在任何机器上都能运行的话，还需要将其推送到镜像仓库。本文不会介绍如何构建一个私有镜像仓库，这个知识点在后面学习到。

这里我们使用阿里云的镜像仓库进行演示，当然你也可以将镜像推送到Docker Hub (<http://hub.docker.com/>) 。

## 1.创建个人镜像仓库

### 创建命名空间

首先进入阿里云的容器镜像服务页面



## 创建镜像仓库

## 创建镜像仓库



1

仓库信息

2

代码源

地域

华东2（上海）

\* 命名空间

david-ns01

选择刚才创建的命名空间

\* 仓库名称

test1

5/64

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末位）

仓库类型

☐ 公开 ☒ 私有

\* 摘要

测试

2/100

长度最长100个字符

描述信息

支持Markdown格式

## 创建镜像仓库



✓

仓库信息

2

代码源

代码源

云Code

GitHub

Bitbucket

私有GitLab

本地仓库

您可以通过命令行推送镜像到镜像仓库。

上一步

创建镜像仓库

取消

## 管理镜像仓库

## ← 默认实例

所在地域 华东2（上海）

概览

仓库管理

镜像仓库

命名空间

代码源

访问凭证

创建镜像仓库

全部命名空间

仓库名称

Q

C

仓库名称	命名空间	仓库状态	仓库类型	仓库地址	创建时间	操作
		✓ 正常	私有	...	2020-06-15 16:39:35	管理   删除
test1	david-ns01	✓ 正常	私有	...	2020-11-05 10:32:34	管理   删除

选择“管理”链接，跳转到如下页面：

## ← test1

基本信息

仓库授权

触发器

镜像版本

镜像同步

基本信息

编辑

仓库名称	test1
仓库地域	华东2（上海）
仓库类型	私有
代码仓库	无

公网地址 ?

专有网络 ?

经典网络 ?

摘要 ?

操作指南 镜像描述

## 1. 登录阿里云Docker Registry

```
$ sudo docker login --username=david**** registry.cn-shanghai.aliyuncs.com
```

用于登录的用户名为阿里云账号全名，密码为开通服务时设置的密码。

您可以在访问凭证页面修改凭证密码。

## 2. 从Registry中拉取镜像

```
$ sudo docker pull registry.cn-shanghai.aliyuncs.com/david-ns01/test1:[镜像版本号]
```

## 3. 将镜像推送到Registry

```
$ sudo docker login --username=david**** registry.cn-shanghai.aliyuncs.com
$ sudo docker tag [ImageId] registry.cn-shanghai.aliyuncs.com/david-ns01/test1:[镜像版本号]
$ sudo docker push registry.cn-shanghai.aliyuncs.com/david-ns01/test1:[镜像版本号]
```

请根据实际镜像信息替换示例中的[ImageId]和[镜像版本号]参数。

## 4. 选择合适的镜像仓库地址

可以看到阿里云已经为我们生成好了推送镜像的命令：

```
$ sudo docker login --username=david**** registry.cn-shanghai.aliyuncs.com
```

```
$ sudo docker tag [ImageId] registry.cn-shanghai.aliyuncs.com/david-ns01/test1:[镜像版本号]
```

```
$ sudo docker push registry.cn-shanghai.aliyuncs.com/david-ns01/test1:[镜像版本号]
```

## 2.推送镜像

现在将我们之前制作好的test1:latest镜像推送到镜像仓库

### 添加标签

先按阿里云的规范给test1:latest镜像添加一个新的tag:

```
sudo docker tag test1 registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
```

```
[root@dhr-demo ~]# sudo docker tag test1 registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
[root@dhr-demo ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
test1	latest	773641d7cb99	10 hours ago	660MB
registry.cn-shanghai.aliyuncs.com/david-ns01/test1	1.0	773641d7cb99	10 hours ago	660MB
netto-world	latest	bf756fb1ae65	10 months ago	13.3kB
node	7	d9aed20b68a4	3 years ago	660MB

```
[root@dhr-demo ~]#
```

该命令会给同一个镜像再创建一个标签。通过docker images命令可以看到test1和registry.cn-shanghai.aliyuncs.com/david-ns01/test1两个标签指向的是同一个镜像ID。

### 登录镜像仓库

如果镜像仓库不是公开的话，还得先登录

```
sudo docker login --username=david*** registry.cn-shanghai.aliyuncs.com
```

```
[root@dhr-demo ~]# sudo docker login --username=david registry.cn-shanghai.aliyuncs.com
Password:
Login Succeeded
[root@dhr-demo ~]#
```

### 推送镜像

```
sudo docker push registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
```

```
[root@dhr-demo ~]# sudo docker push registry.cn-shanghai.aliyuncs.com/david-ns01/test1:1.0
The push refers to repository [registry.cn-shanghai.aliyuncs.com/david-ns01/test1]
955c1ac2b2e4: Pushed
ab90d83fa34a: Pushed
8ee318e54723: Pushed
e6695624484e: Pushed
da59b99bbd3b: Pushed
5616a6292c16: Pushed
f3ed6cb59ab0: Pushed
654f45ecb7e3: Pushed
2c40c66f7667: Pushed
1.0: digest: sha256:672d7b55833edefd0fbc6bc7c52cfd16146b3ef87147d39bce5327de4b6e2ee9 size: 2213
[root@dhr-demo ~]#
```

至此，我们已经成功的将一个镜像推送到仓库中，可以在如下页面看到刚刚推送的镜像：

阿里云

搜索文档、控制台、API、解决方案和资源

费用工单备案企业支持官网

简体

容器镜像服务 / 实例列表 / 镜像仓库 / 镜像版本

← test1

华东2（上海） | 私有 | 本地仓库 | ✓ 正常

部署应用

基本信息

仓库授权

触发器

镜像版本

镜像同步

版本	镜像ID	状态	Digest	镜像大小	最后更新时间	操作
1.0	773641d7cb...	✓ 正常	672d7b55833edefd0fbc6bc7c 52cfd16146b3ef87147d39bce 5327de4b6e2ee9	251.447 MB	2020-11-05 11:01:26	<a href="#">安全扫描</a>   <a href="#">层信息</a>   <a href="#">同步</a>   <a href="#">删除</a>

< 1 >

推送成功后，这个镜像就可以被任何人使用了。以后应用每次都会运行在相同的环境中。如果在你的本机上能正常运行，那么在其他Linux机器上也能正常运行。完全不用担心宿主机是否安装了Node.js。事实上，即使安装了Node.js，应用程序也不会用到它，因为应用程序使用的是镜像里安装的Node.js。