



# 容器镜像服务

用户使用指南

天翼云科技有限公司

## 目录

容器镜像服务.....	1
目录	2
1 产品介绍.....	4
1.1 产品定义 .....	4
1.2 产品功能 .....	4
1.3 应用场景 .....	5
1.4 产品优势 .....	5
1.5 使用限制 .....	5
2 购买指南.....	6
2.1 价格6	
3 快速入门.....	7
3.1 上传并使用私有镜像 .....	7
4 操作指南.....	11
4.1 创建仓库 .....	11
4.2 客户端上传私有镜像 .....	12
4.3 删除私有镜像.....	14
4.4 删除镜像仓库.....	15
5 常见问题.....	17
5.1 通用类 .....	17
5.1.1 Q: 什么是容器镜像服务? .....	17

5.1.2 Q: 什么是项目 .....	17
5.1.3 Q: 容器镜像服务是否收费? .....	17
5.1.4 Q: 容器镜像服务最多能存储多少个镜像? .....	17
5.1.5 Q: 容器镜像服务的配额是多少? .....	17
5.1.6 Q: 为什么创建项目时, 提示项目已存在? .....	17
5.1.7 Q: 容器镜像服务的带宽多大? .....	18
5.1.8 Q: 如何在容器引擎中使用? .....	18
5.1.9 Q: 如何制作容器镜像? .....	18
5.1.10 Q: 如何制作镜像压缩包? .....	23
<b>5.2 镜像管理类.....</b>	<b>24</b>
5.2.1 Q: 为什么使用客户端上传镜像失败? .....	24
5.2.2 Q: 为什么通过客户端和docker images看到的镜像大小不一样? .....	24
5.2.3 Q: docker push将镜像推送到SWR使用的什么协议? .....	24
5.2.4 Q: 如何通过页面下载容器镜像? .....	24
5.2.5 Q: docker pull下载的镜像存放在什么地方? 如何拷贝? .....	24

# 1 产品介绍

## 1.1 产品定义

容器镜像服务(Software Repository for Container, 简称SWR) 是支持容器镜像生命周期管理的服务, 提供简单易用、安全可靠的容器镜像管理功能, 帮助用户快速部署容器化服务。

容器镜像服务可配合容器引擎CCE、容器实例CCI使用, 也可单独作为容器镜像仓库使用。

名词释义:

**项目:** 项目定义了属性与镜像存储空间。依据属性划分, 用户可以创建1个属性为公有的项目, 以及1个属性为私有的项目, 在项目内创建的镜像仓库会继承项目的属性, 在仓库下的镜像属性与镜像仓库相同。同时项目还定义了您存储镜像的空间范围, 如果超出空间限制, 将会上传失败。

**镜像仓库 (Repository):** 提供Docker容器镜像管理功能, 用户在创建容器应用前, 需要将应用所需的镜像上传到镜像仓库。Docker镜像是一个模板, 用于创建Docker容器。Docker提供了一个简单的机制来创建新的镜像或更新已有镜像。

**仓库属性:** 属性分为公有和私有两种。公有: 任何租户、用户均可以下载。私有: 仅当前租户或租户下的用户可用。

## 1.2 产品功能

### 1、提供多种官方镜像

镜像中心提供tomcat、httpd、mysql、nginx等多种常用官方Docker镜像, 支持用户快速部署应用, 搭建业务。

### 2、支持容器镜像仓库的管理, 包括创建、设置属性与删除。

支持用户管理镜像仓库。创建镜像仓库的过程由创建项目与创建仓库两步完成, 首先创建项目并完成属性设置(公有或私有), 在项目下创建仓库时继承该属性。

### 3、支持私有镜像管理, 包括上传和下载

用户可从将本地私有镜像或容器引擎上的镜像上传至镜像仓库中, 以创建容器应用。

镜像仓库不扫描用户上传的镜像, 不负责对用户上传的镜像进行安全性验证。上传的镜像中请不要包含未加密的口令, 密码等隐私信息, 以避免隐私泄露。下载公有镜像时, 应确定数据来自于可信的仓库源, 以避免下载到恶意软件。

如果使用自定义镜像, 请确保镜像来源可信, 不在容器镜像内安装不必要的软件, 在升级时使用安全补丁升级镜像。使用第三方镜像, 造成的后果(例如, 环境不可用)用户需自己承担。

## 1.3 应用场景

容器镜像服务为用户提供了云上的容器镜像管理服务，本服务主要搭配容器引擎CCE、容器实例CCI等产品使用，用于容器引擎创建应用和升级应用过程。

## 1.4 产品优势

### 简单易用

无需自行搭建和运维，即可快速推送拉取容器镜像。

容器镜像服务的管理控制台简单易用，镜像的全生命周期管理操作便利。

### 安全可靠

容器镜像服务遵循HTTPS协议保障镜像安全传输，提供帐号间、帐号内多种安全隔离机制，确保用户数据访问的安全。

容器镜像服务依托专业存储服务，确保镜像存储更可靠。

### 一键部署

容器镜像服务提供容器引擎CCE、容器实例CCI无缝对接，一键式部署容器应用

## 1.5 使用限制

### 配额

容器镜像服务对单个用户创建的项目数量、项目内镜像存储空间进行了限制。如下表所示。

资源类型	释义	默认配额	可调配额
项目	单用户可创建的项目数量	10G存储空间	无法调整
镜像仓库	单个项目可创建的镜像仓库数量	数量无限制	不涉及
镜像	单个镜像仓库可上传的镜像数量	数量无限制 全量镜像空间不超过项目存储空间	不涉及

## 2 购买指南

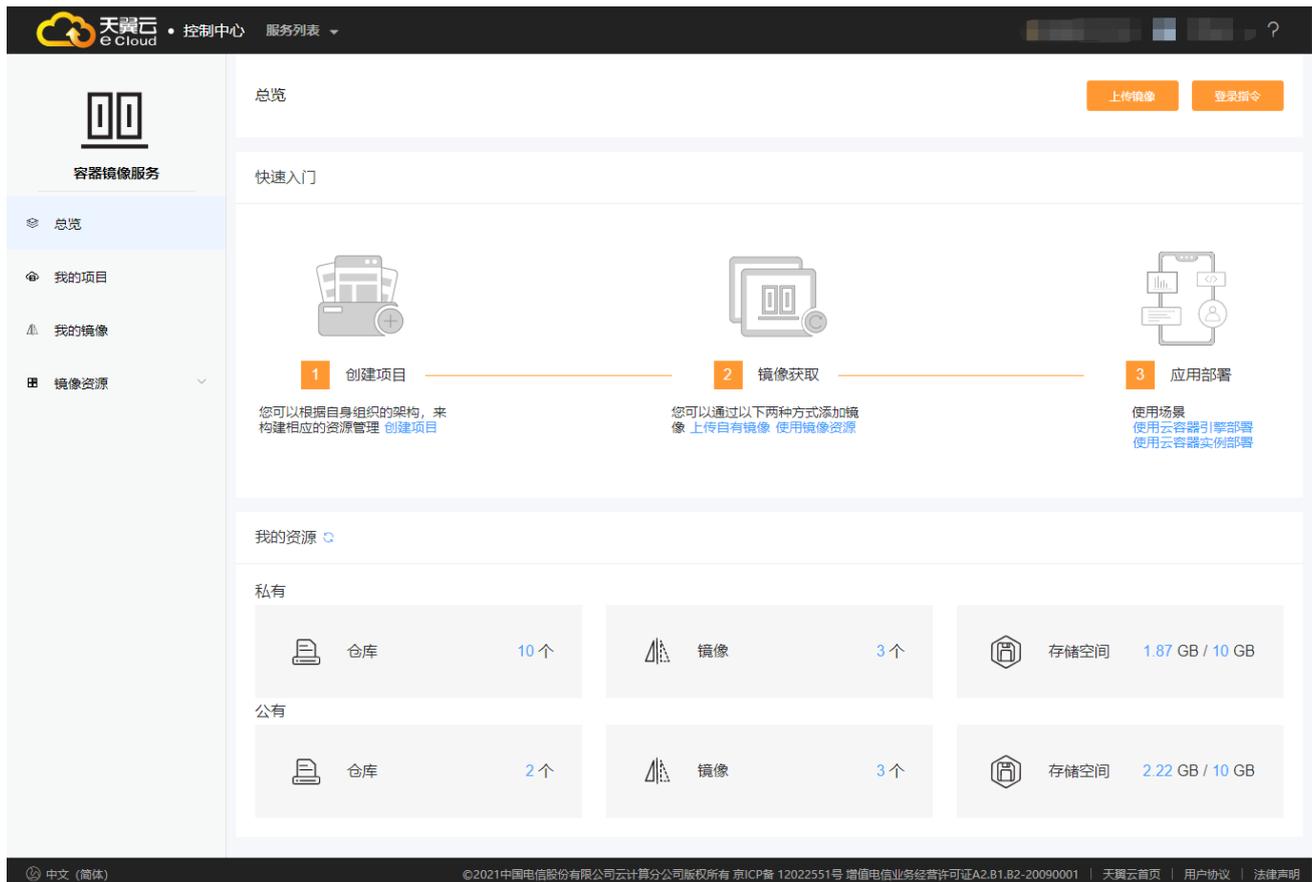
### 2.1 价格

目前容器镜像服务免费

## 3 快速入门

### 3.1 上传并使用私有镜像

※注：在使用本产品之前，您需要先注册天翼云门户的账号。获取账号后，通过控制台总览，找到容器镜像服务，点击进入容器镜像服务控制台首页，开启您下面步骤的操作：



#### 1、创建项目

在控制台首页，点击【创建项目】，进入创建项目界面。此过程需要输入项目名称，并设置项目的属性，随后完成创建。其中，项目属性包括公有或私有，这决定着在这个项目内镜像仓库的属性，决定着您上传的私有镜像可用范围。

私有：镜像只有自己可在创建应用过程使用

公有：镜像全网容器用户均可在创建应用过程使用



天翼云 e Cloud • 控制中心 服务列表

### 创建项目

容器镜像服务

总览

我的项目

我的镜像

镜像资源

创建

取

公有

私有

基本信息

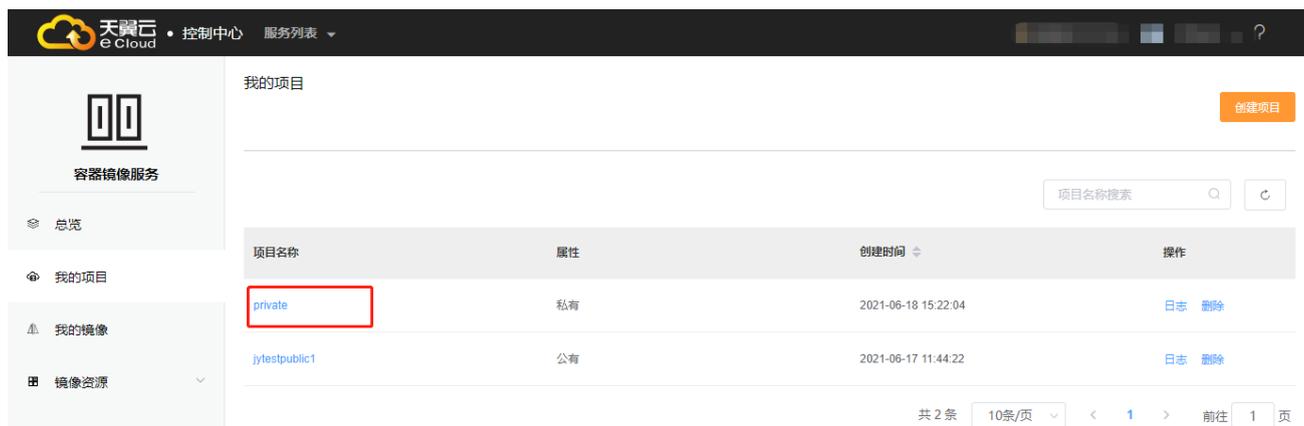
\* 项目名称: private

\* 属性: 请选择属性

※注：当前用户可创建属性为公有的项目与私有的项目各1个，无法调整配额。

## 2、创建镜像仓库

点击左侧导航栏**我的项目**，点击已创建好的项目，点击**创建仓库**，输入仓库名称，完成创建



天翼云 e Cloud • 控制中心 服务列表

### 我的项目

创建项目

项目名称搜索

项目名称	属性	创建时间	操作
private	私有	2021-06-18 15:22:04	日志 删除
jylestpublic1	公有	2021-06-17 11:44:22	日志 删除

共 2 条 10条/页 < 1 > 前往 1 页



天翼云 e Cloud • 控制中心 服务列表

### 我的仓库

创建仓库

仓库名称搜索

仓库名称	属性	创建时间	操作
privatepri11	私有	2021-06-21 10:48:39	删除

天翼云 e Cloud • 控制中心 服务列表 ▾



容器镜像服务

- ☰ 总览
- 📁 我的项目
- 📁 我的镜像
- 📁 镜像资源 ▾

### 创建镜像仓库

---

基本信息

\* 仓库名称:

创建
取消

### 3、上传镜像

点击已创建好的镜像仓库，点击**上传镜像**，提示上传镜像的操作步骤。其中，点击查看密码，可查看镜像仓库服务器的密码。您按照说明中的操作步骤即可完成镜像上传（详细示例请见4.2客户端上传私有镜像）

天翼云 e Cloud • 控制中心 服务列表 ▾



容器镜像服务

- ☰ 总览
- 📁 我的项目
- 📁 我的镜像

### 我的仓库

创建仓库

Q
C

仓库名称	属性	创建时间	操作
private/demo	私有	2021-08-02 14:46:41	<a href="#">删除</a>

天翼云 e Cloud • 控制中心 服务列表 ▾



容器镜像服务

- ☰ 总览
- 📁 我的项目
- 📁 我的镜像

### 我的镜像

上传镜像

private ▾
 private/demo ▾
 
Q
C

镜像名称	项目	摘要数量	更新时间	创建时间	操作
暂无数据					

## 上传镜像


**步骤 1: 登录镜像仓库服务器**

```
docker login -u {username} -p {pwd} zhongwei2.ccr.ctyun.cn
```

 用户名: 

 密码: \*\*\*\*\* [查看密码](#)
**步骤 2: 标记本地镜像**

```
docker tag { 镜像ID } zhongwei2.ccr.ctyun.cn/{ 项目名称 }/{ 仓库名 }/{ 镜像名 }:{ 标签名 }
```

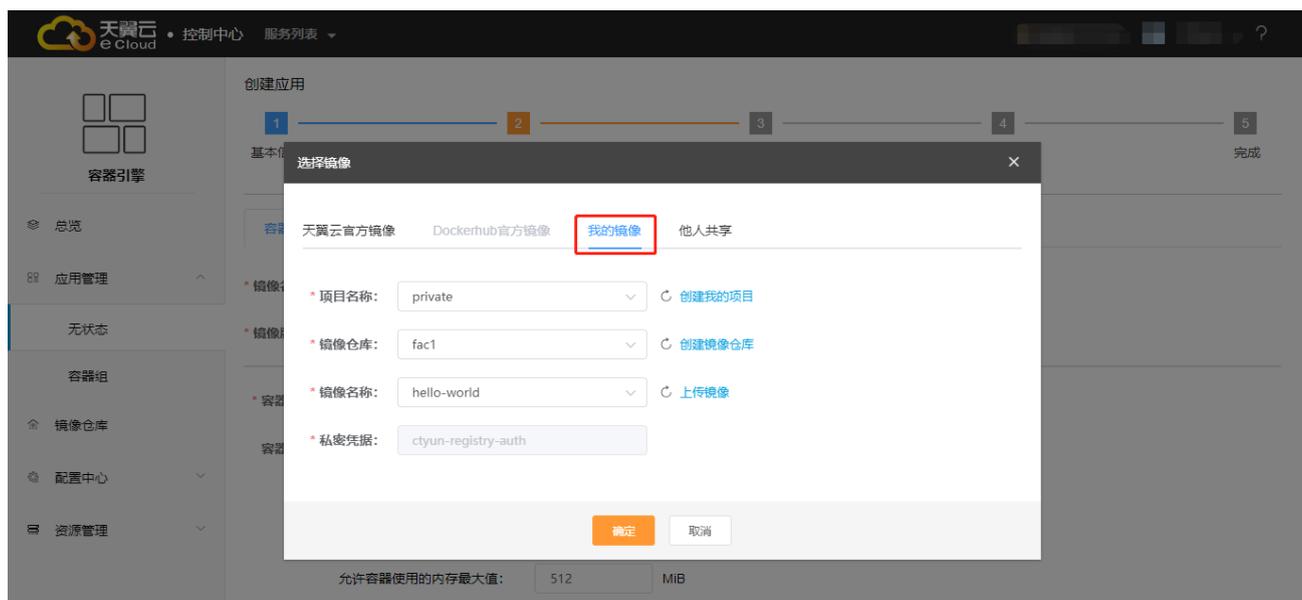
**步骤 3: 推送镜像到仓库**

```
docker push zhongwei2.ccr.ctyun.cn/{ 项目名称 }/{ 仓库名 }/{ 镜像名 }:{ 标签名 }
```

关闭

**4、拉取镜像创建或升级应用**

在容器引擎创建应用或应用升级的过程，通过项目名称、镜像仓库，可以选择到上传的私有镜像。



※注：容器引擎使用说明，请见<https://www.ctyun.cn/h5/help2/10000068>

## 4 操作指南

### 4.1 创建仓库

创建镜像仓库的过程由**创建项目**与**创建仓库**两步完成。

#### 1、创建项目

在控制台首页，点击**创建项目**，进入创建项目界面。此过程需要输入项目名称，并设置项目的属性，随后完成创建。其中，项目属性包括公有或私有，决定着在这个项目内镜像仓库的属性，决定着您上传的私有镜像可用范围。

私有：镜像只有自己可在创建应用过程使用

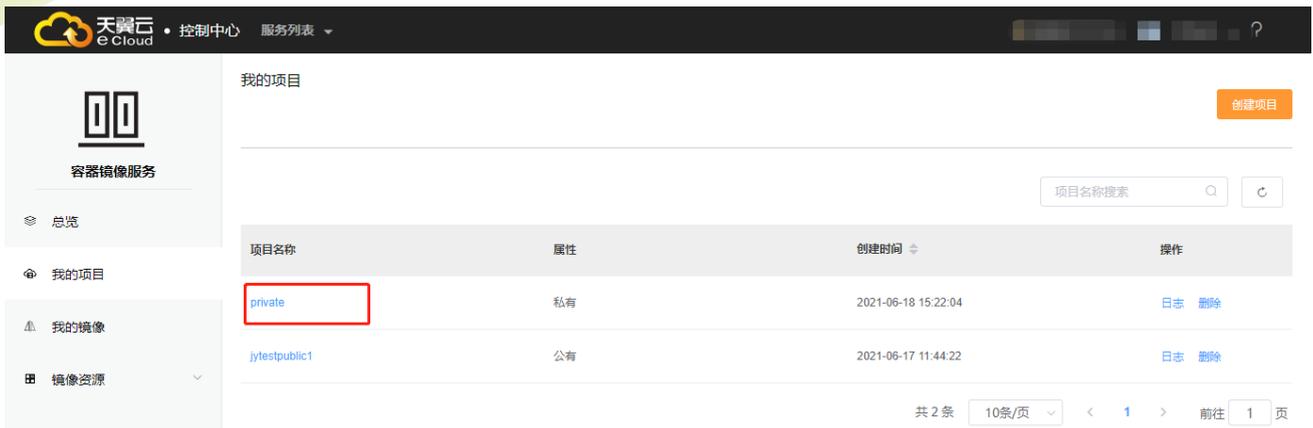
公有：镜像全网容器用户均可在创建应用过程使用



※注：当前用户可创建属性为公有的项目与私有的项目各1个，无法调整配额。

#### 2、创建镜像仓库

点击左侧导航栏**我的项目**，点击已创建好的项目，点击**创建仓库**，输入仓库名称，完成创建



天翼云 eCloud · 控制中心 服务列表

我的项目 创建项目

项目名称搜索

项目名称	属性	创建时间	操作
private	私有	2021-06-18 15:22:04	日志 删除
jytestpublic1	公有	2021-06-17 11:44:22	日志 删除

共 2 条 10条/页 < 1 > 前往 1 页

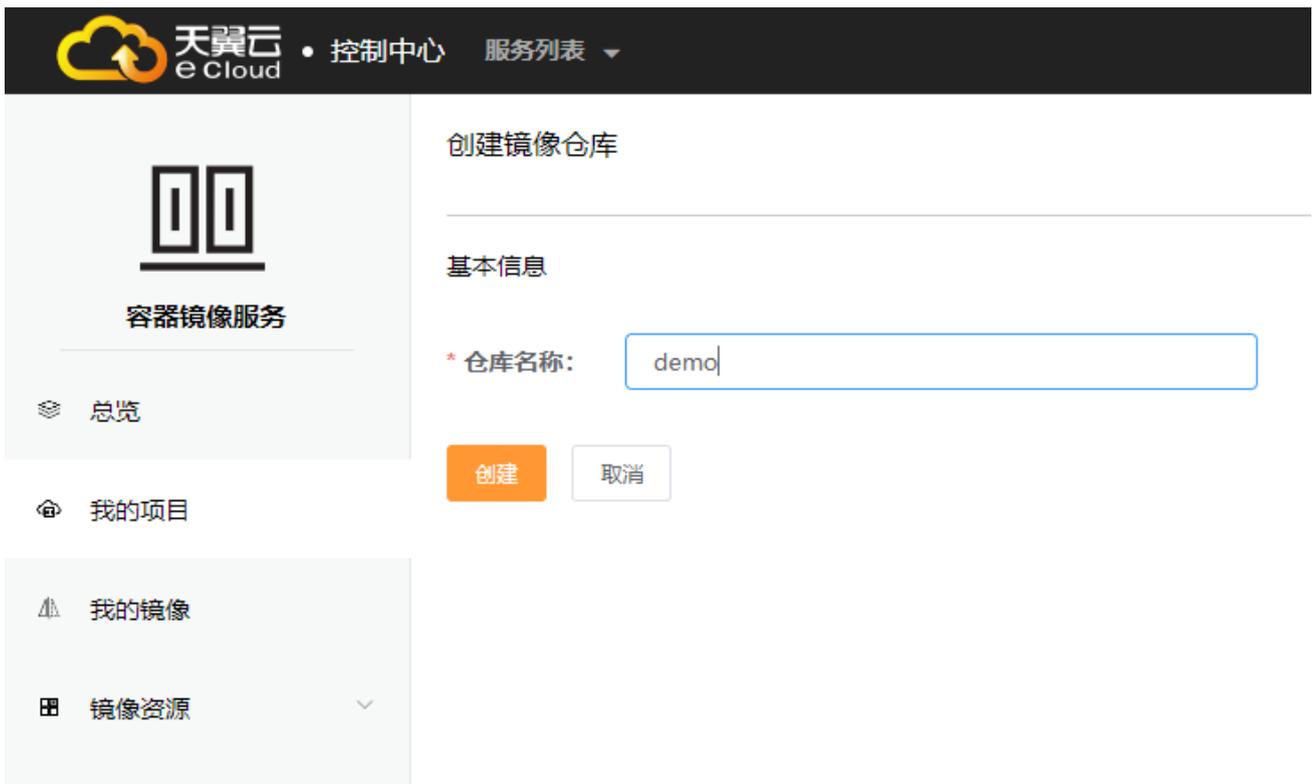


天翼云 eCloud · 控制中心 服务列表

我的仓库 创建仓库

仓库名称搜索

仓库名称	属性	创建时间	操作
privatepri11	私有	2021-06-21 10:48:39	删除



天翼云 eCloud · 控制中心 服务列表

容器镜像服务

创建镜像仓库

基本信息

\* 仓库名称:

创建 取消

※注：镜像仓库的属性（公有/私有）与所在项目保持一致

## 4.2 客户端上传私有镜像

通过客户端上传私有镜像，指的是使用命令将镜像上传到容器镜像服务的镜像仓库。本章节选择宁夏-

中卫2资源池，以nginx:1.10镜像为例，介绍如何通过客户端上传私有镜像。

### 前置工作

- 1、已安装Docker客户端，并确定Docker服务已启动，可联通公网；
- 2、请确保镜像的正确性，能够成功后台启动；
- 3、已创建名为nginx的容器镜像仓库，操作过程请参见 3.1 创建容器镜像仓库。

### 操作步骤：

- 1、以下步骤讲解通过客户端Push镜像到镜像仓库；

#### 1) 登录镜像仓库服务器：

- A、点击**上传镜像**，获取登录指令，用户名可直接获取，密码可通过查看密码获取：



- B、按照步骤1格式所示，输入docker login命令；

#### 2) 标记nginx:1.10镜像（步骤2）：

- A、使用命令获得镜像名称：`docker images`

- B、标记nginx:1.10镜像：`docker tag {镜像ID} zhongwei2.ccr.ctyun.cn/{项目名称}/{仓库名}/{镜像名}:{标签名}`

**docker tag nginx zhongwei2.ccr.ctyun.cn/private/nginx/nginx:1.10**

其中，如**zhongwei2.ccr.ctyun.cn**这部分，各资源池的外网地址均可在上传镜像步骤中查看到，**1.10** 为版本号。

#### 3) 推送镜像至镜像仓库（步骤3）：

**docker push zhongwei2.ccr.ctyun.cn/{项目名称}/{仓库名}/{镜像名}:{标签名}** 样例如下：

**docker push zhongwei2.ccr.ctyun.cn/private/nginx/nginx:1.10** 终端显示如下信息，表明 push 镜像成功。

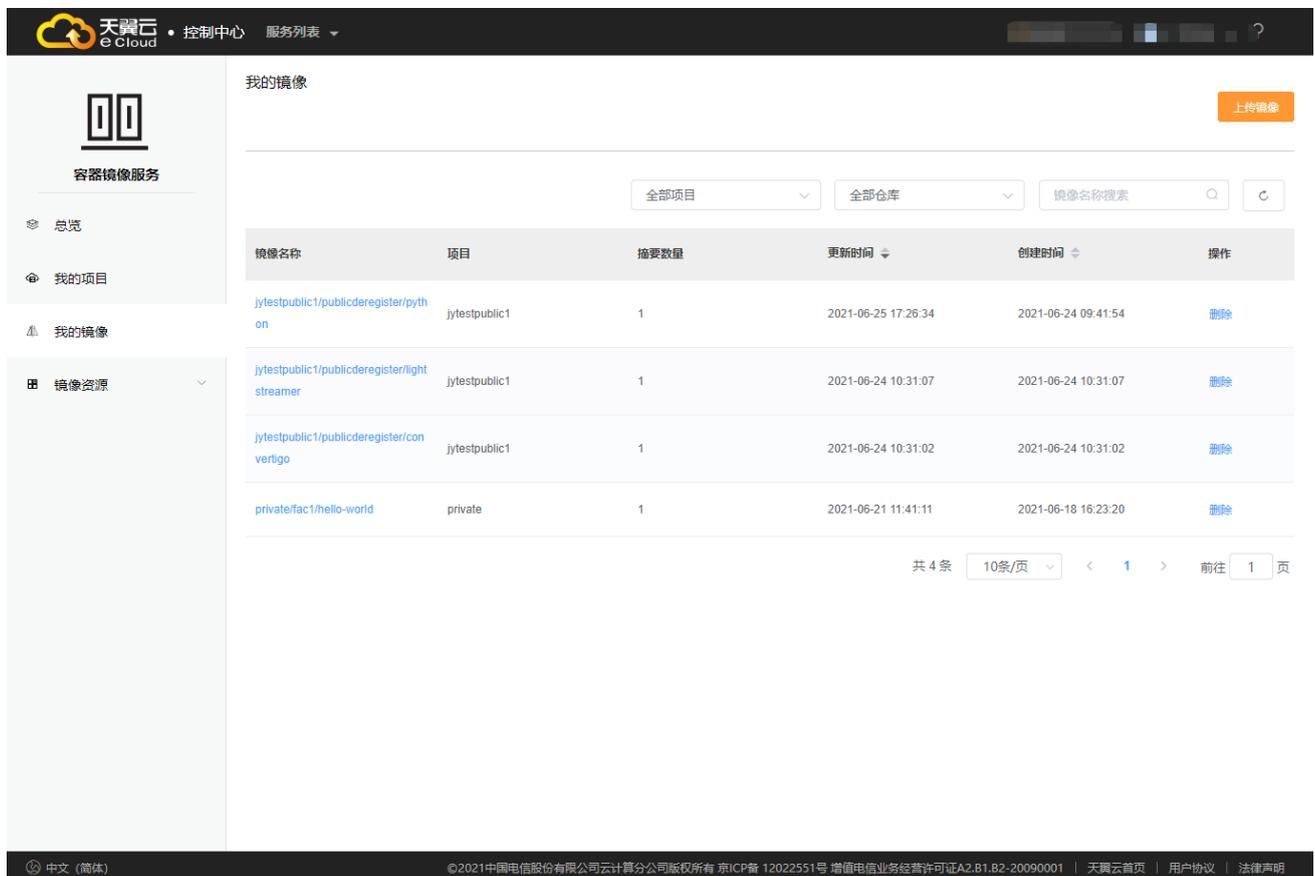
```
6d6b9812c8ae: Pushed
695da0025de6: Pushed
fe4c16cbf7a4: Pushed
1.10: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63
size: 948
```

返回系统，在容器镜像仓库的“镜像详情”页面，执行刷新操作后可查看到对应的镜像信息。

## 4.3 删除私有镜像

※注：删除的镜像无法找回，请您谨慎操作。

- 1、点击**我的镜像**，可看到全量私有镜像。对需要删除的镜像，点击操作列的删除。如有必要，可在上方项目、仓库内进行筛选，或按镜像名称搜索，以便快速定位需要删除的镜像。



天翼云 eCloud 控制中心 服务列表

容器镜像服务

我的镜像

上传镜像

全部项目 全部仓库 镜像名称搜索

镜像名称	项目	摘要数量	更新时间	创建时间	操作
jltestpublic1/publicderegister/python	jltestpublic1	1	2021-06-25 17:26:34	2021-06-24 09:41:54	删除
jltestpublic1/publicderegister/lightstreamer	jltestpublic1	1	2021-06-24 10:31:07	2021-06-24 10:31:07	删除
jltestpublic1/publicderegister/convertigo	jltestpublic1	1	2021-06-24 10:31:02	2021-06-24 10:31:02	删除
private/fac1/hello-world	private	1	2021-06-21 11:41:11	2021-06-18 16:23:20	删除

共 4 条 10条/页 < 1 > 前往 1 页

中文 (简体) ©2021 中国电信股份有限公司云计算分公司版权所有 京ICP备 12022551号 增值电信业务经营许可证A2.B1.B2-20090001 | 天翼云首页 | 用户协议 | 法律声明

- 2、弹出删除确认提示框，点击确定，完成删除。



## 4.4 删除镜像仓库

※注：删除镜像仓库前，需要您将该镜像仓库下的镜像全部清除。

1、点击我的项目，点击待删除镜像仓库所在的项目



2、选择待删除的镜像仓库，在操作列点击删除



3、弹出删除确认提示框，点击确定，完成删除。

删除

✕

确定要将镜像仓库private/pri11删除么？

确定

取消

登录云容器引擎控制台，单击左侧导航栏的【镜像仓库】；

## 5 常见问题

### 5.1 通用类

#### 5.1.1 Q: 什么是容器镜像服务?

Q: 什么是容器镜像服务?

A: 容器镜像服务（SoftWare Repository for Container）是一种支持容器镜像全生命周期管理的服务，提供简单易用、安全可靠的镜像管理功能，帮助用户快速部署容器化服务。

#### 5.1.2 Q: 什么是项目

Q: 什么是项目?

A: 项目定义了属性与镜像存储空间。依据属性划分，用户可以创建1个属性为公有的项目，以及1个属性为私有的项目，在项目内创建的镜像仓库会继承项目的属性，在仓库下的镜像属性与镜像仓库相同。同时项目还定义了您存储镜像的空间范围，如果超出空间限制，将会上传失败。

#### 5.1.3 Q: 容器镜像服务是否收费?

Q: 容器镜像仓库服务是否收费?

A: 容器镜像服务暂不收费，可以免费使用。

#### 5.1.4 Q: 容器镜像服务最多能存储多少个镜像?

Q: 容器镜像服务最多能存储多少个镜像?

A: 容器镜像服务对存储的镜像数量没有限制，但是镜像所占用空间之和无法超过镜像所在项目的总空间，您可以根据需要合理安排项目空间的使用。

#### 5.1.5 Q: 容器镜像服务的配额是多少?

Q: 容器镜像服务的配额是多少?

A: 容器镜像服务仅限制用户创建项目的数量，用户最多可创建属性为公有的项目1个，属性为私有的项目1个，且无法调整。镜像仓库与镜像均无数量限制，镜像的所占空间之和不超过项目的存储空间。

#### 5.1.6 Q: 为什么创建项目时，提示项目已存在?

**Q: 为什么创建项目时，提示项目已存在？**

A: 容器镜像服务中项目名称是所有账号共用的，比如A用户创建了“projectA”，B用户就没法使用“projectA”。因此创建项目时如果提示项目已存在，可能项目名称已被其他用户使用，请重新设置一个项目名称。

### 5.1.7 Q: 容器镜像服务的带宽多大？

**Q: 容器镜像服务的带宽多大？**

A: 容器镜像服务的带宽会根据用户使用情况动态变化。

### 5.1.8 Q: 如何在容器引擎中使用？

**Q: 如何在容器引擎中使用？**

A: 用户再容器引擎产品中创建应用与应用升级的过程中，选择镜像时会在**我的镜像**中看到上传过的私有镜像。

容器引擎创建应用操作说明：<https://www.ctyun.cn/h5/help2/10000068/10001129>

容器引擎应用升级操作说明：<https://www.ctyun.cn/h5/help2/10000068/10330246>

### 5.1.9 Q: 如何制作容器镜像？

**Q: 如何制作容器镜像？**

A: 自己制作容器镜像，主要有两种方法：

- 制作快照方式获得镜像（偶尔制作的镜像）：在基础镜像上，比如Ubuntu，先登录镜像系统并安装容器引擎软件，然后整体制作快照。
- Dockerfile方式构建镜像（经常更新的镜像）：将软件安装的流程写成DockerFile，使用Docker build构建成容器镜像。
- 方法一：制作快照方式获得镜像

如果后续镜像没有变化，可采用方法一制作镜像。



具体操作如下：

1. 找一台主机，安装容器引擎软件。
2. 启动一个空白的基础容器，并进入容器。

例如：启动一个CentOS的容器。

```
docker run -it centos
```

3. 执行安装任务。

```
yum install XXX
```

```
git clone https://github.com/lh3/bwa.git
```

```
cd bwa;make
```

说明：

请预先安装好Git，并检查本机是否有ssh key设置。

4. 输入exit退出容器。
5. 制作快照。

```
docker commit -m "xx" -a "tsj" container-id tsj/image:tag
```

- o -a：提交的镜像作者。
- o container-id：操作步骤2中的容器id。可以使用 `docker ps -a` 查询得到容器id。
- o -m：提交时的说明文字。
- o tsj/image:tag：仓库名/镜像名:TAG名。

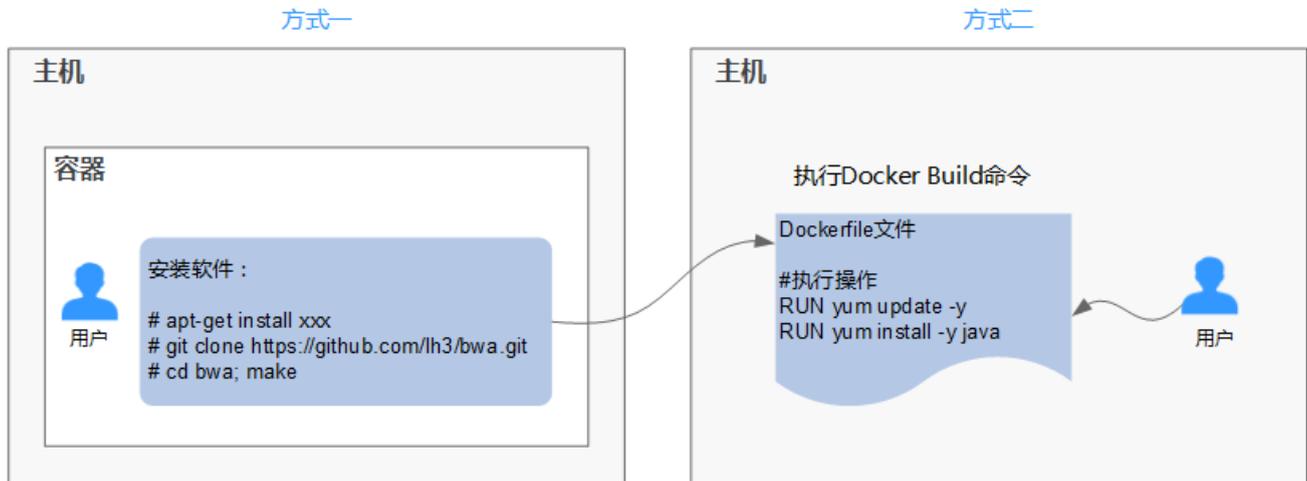
6. 执行docker images可以查看到制作完成的容器镜像。

- 方法二：使用Dockerfile方式构建

如果后续镜像经常变更（例如某个软件更新版本），则需要采用方法二制作镜像。若仍采用方法一制作镜

像，则每次变更都需要重新执行方法一的命令，操作过程比较繁琐，所以建议使用自动化制作镜像的方法。

其实就是将方法一制作镜像的方法，用文件方式写出来（文件名为DockerFile）。然后执行：`docker build -t tsj/image:tag.命令`（命令中“.”表示DockerFile文件的路径），自动完成镜像制作。



简单的DockerFile示例:

说明:

如果依赖外部网络，请搭建网络环境，并保证网络可用。

`#Version 1.0.1`

`FROM centos:latest`

`MAINTAINER ***u "****u@huaweicloud.com"`

`#设置root用户为后续命令的执行者`

`USER root`

`#执行操作`

`RUN yum update -y`

`RUN yum install -y java`

`#使用&&拼接命令`

`RUN touch test.txt && echo "abc" >>abc.txt`

`#对外暴露端口`

`EXPOSE 80 8080 1038`

`#添加网络文件`

```
ADD https://www.baidu.com/img/bd_logo1.png /opt/
```

```
#设置环境变量
```

```
ENV WEBAPP_PORT=9090
```

```
#设置工作目录
```

```
WORKDIR /opt/
```

```
#设置启动命令
```

```
ENTRYPOINT ["ls"]
```

```
#设置启动参数
```

```
CMD ["-a", "-l"]
```

```
#设置卷
```

```
VOLUME ["/data", "/var/www"]
```

```
#设置子镜像的触发操作
```

```
ONBUILD ADD ./app/src
```

```
ONBUILD RUN echo "on build excuted" >> onbuild.txt
```

- **DockerFile基本语法**

- **FROM:**

指定待扩展的父级镜像（基础镜像）。除注释之外，文件开头必须是一个**FROM**指令，后面的指令便在这个父级镜像的环境中运行，直到遇到下一个**FROM**指令。通过添加多个**FROM**命令，可以在同一个Dockerfile文件中创建多个镜像。

- **MAINTAINER:**

声明创建镜像的作者信息：用户名、邮箱，非必须参数。

- **RUN:**

修改镜像的命令，常用来安装库、安装程序以及配置程序。一条**RUN**指令执行完毕后，会在当前镜像上创建一个新的镜像层，接下来对的指令会在新的镜像上继续执行。**RUN** 语句有两种形式：

- o **RUN yum update:** 在/bin/sh路径中执行的指令命令。

- o **RUN ["yum", "update"]:** 直接使用系统调用exec来执行。

- o **RUN yum update && yum install nginx:** 使用&&符号将多条命令连接在同一条**RUN**语句中。

- **EXPOSE:**

指明容器内进程对外开放的端口，多个端口之间使用空格隔开。

运行容器时，通过设置参数-P（大写）即可将EXPOSE里所指定的端口映射到主机上其他的随机端口，其他容器或主机可以通过映射后的端口与此容器通信。

您可以通过设置参数-p（小写）将Dockerfile中EXPOSE中没有列出的端口设置成公开。

- **ADD:**

向新镜像中添加文件，这个文件可以是一个主机文件，也可以是一个网络文件，也可以是一个文件夹。

- o 第一个参数：源文件（夹）。

如果是相对路径，必须是相对于Dockerfile所在目录的相对路径。

如果是URL，会将文件先下载下来，然后再添加到镜像里。

- o 第二个参数：目标路径。

如果源文件是主机上的zip或者tar形式的压缩文件，容器引擎会先解压缩，然后将文件添加到镜像的指定位置。

如果源文件是一个通过URL指定的网络压缩文件，则不会解压。

- **VOLUME:**

在镜像里创建一个指定路径(文件或文件夹)的挂载点，这个容器可以来自主机或者其它容器。多个容器可以通过同一个挂载点共享数据，即便其中一个容器已经停止，挂载点也仍可以访问。

- **WORKDIR:**

为接下来执行的指令指定一个新的工作目录，这个目录可以是绝对目录，也可以是相对目录。根据需要，WORKDIR可以被多次指定。当启动一个容器时，最后一条WORKDIR指令所指的目录将作为容器运行的当前工作目录。

- **ENV:**

设置容器运行的环境变量。在运行容器的时候，通过设置-e参数可以修改这个环境变量值，也可以添加新的环境变量。

例如：

```
docker run -e WEBAPP_PORT=8000 -e WEBAPP_HOST=www.example.com ...
```

- **CMD:**

用来设置启动容器时默认运行的命令。

- **ENTRYPOINT:**

用来指定容器启动时的默认运行的命令。区别在于：运行容器时添加在镜像之后的参数，对ENTRYPOINT是拼接，CMD是覆盖。

- o 若在DockerFile中指定了容器启动时的默认运行命令为ls -l，则运行容器时默认启动命令为 ls -l，例如：

ENTRYPOINT [ "ls", "-l" ]：指定容器启动时的程序及参数为 ls -l 。

docker run centos：当运行centos容器时，默认执行的命令是docker run centos ls -l

docker run centos -a: 当运行centos容器时拼接了-a参数, 则默认运行的命令是docker run centos ls -l -a

o 若在Dockerfile中指定了指定了容器启动时的默认运行命令为--entrypoint, 则在运行容器时若需要替换默认运行命令, 可以通过添加--entrypoint参数来替换Dockerfile中的指定。例如:

```
docker run gutianlangyu/test --entrypoint echo "hello world"
```

- **USER:**

为容器的运行及RUN、CMD、ENTRYPOINT等指令的运行指定用户或UID。

- **ONBUILD:**

触发器指令。构建镜像时, 容器引擎的镜像构建器会将所有的ONBUILD指令指定的命令保存到镜像的元数据中, 这些命令在当前镜像的构建过程中并不会执行。只有新的镜像使用FROM指令指定父镜像为当前镜像时, 才会触发执行。

使用FROM以这个Dockerfile构建出的镜像为父镜像, 构建子镜像时:

```
ONBUILD ADD ./app/src: 自动执行ADD ./app/src
```

### 5.1.10 Q: 如何制作镜像压缩包?

※注: 在本产品不支持上传镜像压缩包, 因此不建议您使用这种方法制作并上传。

使用docker save命令可将容器镜像制作成tar或tar.gz文件压缩包, 具体命令格式如下:

```
docker save [OPTIONS] IMAGE [IMAGE...]
```

OPTIONS说明: --output, -o, 表示导出到文件。

示例:

```
$ docker save nginx:latest > nginx.tar
```

```
$ ls -sh nginx.tar
```

```
108M nginx.tar
```

```
$ docker save php:5-apache > php.tar.gz
```

```
$ ls -sh php.tar.gz
```

```
372M php.tar.gz
```

```
$ docker save --output nginx.tar nginx
```

```
$ ls -sh nginx.tar
```

```
108M nginx.tar
```

```
$ docker save -o nginx-all.tar nginx
```

```
$ docker save -o nginx-latest.tar nginx:latest
```

## 5.2 镜像管理类

### 5.2.1 Q: 为什么使用客户端上传镜像失败?

Q: 为什么使用客户端上传镜像失败?

A: 在使用 **docker push** 命令的过程中（步骤3），可能会出现下列失败情况，下文列出问题现象与解决办法

1.问题现象：提示 **denied: you do not have the permission**

引发原因：push的路径中，镜像服务域名、项目名称、仓库名称存在问题，均会引发此问题。

解决方法：请按步骤1中的用户名和密码，仔细核对登录是否正确

2.问题现象：提示 **tag does not exist: xxxx** 或 **an image does not exist locally with the tag: xxxx**

问题原因：上传的镜像或镜像版本不存在。

解决方法：检查步骤2标记的路径是否存在问题，重新执行步骤2和步骤3。

### 5.2.2 Q: 为什么通过客户端和docker images看到的镜像大小不一样?

Q: 为什么通过客户端和docker images看到的镜像大小不一样?

A: 使用客户端上传的镜像每一层layer进行了tgz压缩，而本地查看的镜像大小是没有经过压缩的，所以大小显示会不一致。

### 5.2.3 Q: docker push将镜像推送到SWR使用的什么协议?

Q: docker push将镜像推送到容器镜像服务使用的什么协议?

A: docker push将镜像推送到容器镜像服务使用的是HTTPS协议。

### 5.2.4 Q: 如何通过页面下载容器镜像?

Q: 如何通过页面下载容器镜像?

A: 目前容器镜像服务界面不支持直接下载镜像压缩包。建议您通过docker客户端使用如下命令下载镜像。

docker pull {节点域名}/[项目名称]/[仓库名称]/[镜像名称:版本名称]，以中卫2为例：**docker pull zhongwei2.ccr.ctyun.cn/[项目名称]/[仓库名称]/[镜像名称:版本名称]**，域名可在上传镜像的说明中获得。

### 5.2.5 Q: docker pull下载的镜像存放在什么地方? 如何拷贝?

Q: docker pull下载的镜像存放在什么地方? 如何拷贝?

A: docker pull将镜像下载到节点本地，您可以通过docker save命令将镜像保存成tar归档文件。

## 修订记录

发布日期	修改说明
2021年10月	首次发布容器镜像服务用户手册