

07 统一镜像管理：镜像仓库介绍

更新时间：2020-08-03 09:39:10



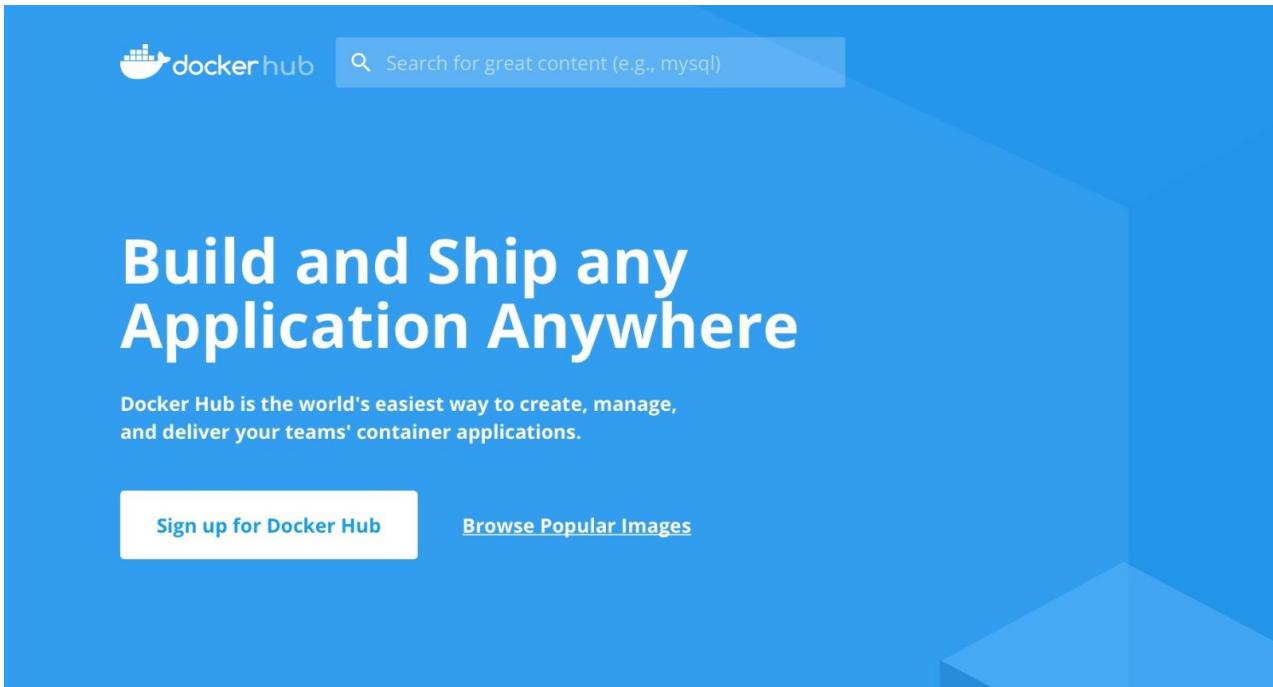
构成我们学习最大障碍的是已知的东西，而不是未知的东西。——贝尔纳

前面简单提到过 Docker 镜像仓库，这里我们再来详细介绍一下 Docker 的镜像仓库中心，也就是 Docker Registry。本文涉及到的 Registry 主要有三个，由于云厂商的 Registry 大同小异，所以下文中我们将主要介绍官方的 Docker Hub 和使用最多的阿里云的 Docker Registry。

- 官方 Docker Registry, Docker Hub;
- 阿里云 Docker Registry;
- 腾讯云 Docker Registry。

1. Docker Hub

Docker Hub 是 Docker 官方提供的 Registry，官网页面如下：



如果我们想要下载 `ubuntu` 镜像，那么在搜索框输入 `ubuntu` 进行搜索就得到下面的结果。

The screenshot shows the Docker Hub search results for the 'ubuntu' image. The page includes the Docker Hub logo, a search bar, and navigation links for Explore, Sign In, Pricing, and Get Started. The main content area displays the 'ubuntu' image card, which features the Ubuntu logo, the name 'ubuntu ☆', and the text 'Docker Official Images'. Below this, a description states: 'Ubuntu is a Debian-based Linux operating system based on free software.' The card also shows a download count of '10M+' and a list of supported tags: 'Container', 'Linux', 'PowerPC 64 LE', 'x86-64', '386', 'ARM 64', 'ARM', 'IBM Z', 'Base Images', 'Operating Systems', and 'Official Image'. On the right side, there is a dropdown menu set to 'Linux - PowerPC 64 LE (latest)', a 'Copy and paste to pull this image' button, and a black button containing the command 'docker pull ubuntu'. Below these buttons is a link 'View Available Tags'. At the bottom of the image card, there are tabs for 'Description' (which is selected), 'Reviews', and 'Tags'.

其中右侧的黑框中的 `docker pull ubuntu` 就是镜像的拉取方式。

Docker Hub 使用起来非常的简单，但是我们一般也只是使用 Docker Hub 来搜索并下载镜像，并不会用来上传存储我们的自定义镜像，这其中的一个主要原因就是网络原因。所以关于 Docker Hub 的镜像上传我们这里就不介绍了。推荐使用云厂商的镜像仓库中心，比如阿里云。

2. 阿里云 Docker Registry

阿里云的 Docker Registry 可以在容器镜像服务 ACR 中找到。容器镜像服务（简称 ACR，下面都简称为 ACR）提供云原生资产的安全托管和全生命周期管理，支持多场景下镜像的高效分发，与容器服务 ACK 无缝集成，打造云原生应用一站式解决方案。

ACR 简化了 **Registry** 的搭建运维工作，支持多地域的镜像托管，并联合容器服务等云产品，打造云上使用 **Docker** 的一体化体验。下面是官方网站上面介绍的 **ACR** 的优势。

ACR 优势

多架构镜像托管支持

- 支持 **Linux**、**Windows**、**ARM** 等多架构容器镜像。

灵活的地域选择

- 用户可以根据自己的业务需求，选择不同的地域创建和删除镜像仓库。
- 每个镜像仓库都提供了公网、内网、VPC 网络下对应的网络地址。

镜像安全扫描

- 支持便捷的镜像安全扫描功能，展示详细的镜像层信息。
- 提供镜像漏洞报告，展示漏洞编号、漏洞等级、修复版本等多维度漏洞信息。

稳定构建服务

- 支持阿里云 **Code**、**GitHub**、**Bitbucket**、自建 **GitLab** 的源代码构建源。
- 支持自动构建，实现源代码变更后自动构建成新 **Docker** 镜像。
- 支持海外构建，实现源代码在海外构建成新 **Docker** 镜像。

云产品间无缝集成

- 整合阿里云 **Code** 及 **CodePipeline**，实现源代码到应用编译测试后，自动构建成新镜像。
- 整合容器服务，实现新镜像构建完成后便捷部署至容器服务集群。

由上面的介绍我们可以看出如果我们最终的容器要运行在阿里云上面，那么阿里云的 **Docker Registry** 绝对是我们最佳的选择。下面我们就来看一下如何使用。

基本介绍

我们直接访问[容器镜像服务控制台](#)，会看到如下的一个页面：

The screenshot shows the Alipay Cloud Container Registry (ACR) service console. On the left, there's a sidebar with tabs for 'Container Registry Service' (selected), 'Default Instance', 'Namespace', 'Authorization Management', 'Code Source', and 'Access Token'. The main area has a header 'Image Repository' and a search bar. Below the header, there's a notice about the enterprise version being released and a promotion for the Tianchi competition. A 'Create Image Repository' button is visible. The main table lists two repositories: one named 'repository' with status 'Normal', type 'Private', and management link; and another with a partially obscured name, status 'Normal', type 'Private', and management link.

我们首先看到右侧的一个 tab:

镜像仓库：比如阿里云上面一个完整的镜像 url 形如 registry.cn-hangzhou.aliyuncs.com/imooc/imooc-demo:v1, 最后面的 imooc-demo 就是镜像仓库。

命名空间：上面的镜像 url 中的 imooc 就是镜像空间。

授权管理：是阿里云账号体系相关的，我们可以暂时不用管。

代码源：构建镜像的代码源，支持阿里云代码管理、Github、Bitbucket、GitLab。除了第一个是阿里云私有的，其他三个都相当于开源的。

The screenshot shows the AliCloud Docker Registry interface. On the left, there's a sidebar with sections like '默认实例' (Default Instances), '代码源' (Code Sources) which is highlighted in blue, '企业版实例' (Enterprise Edition Instances), and '镜像中心' (Image Center). The main content area has four cards: 1. AliCloud Code Management: icon of a code editor, text '阿里云代码管理实现了私有并且安全的代码托管服务，提供目前最流行的分布式版本控制系统Git来有效管理您的项目。' 2. Github: icon of a person, text 'Github 于2008年上线，目前，除了Git代码仓库托管及基本的Web管理界面以外，还提供了订阅、讨论组、文本渲染、在线文件编辑器、协作图谱（报表）、代码片段分享（Gist）等功能。' 3. Bitbucket: icon of a bucket, text 'Bitbucket 是一家源代码托管网站，采用Mercurial和Git作为分布式版本控制系统，同时提供商业计划和免费账户。' 4. GitLab: icon of a person, text 'GitLab 是目前主流的企业内部 Git 代码托管解决方案，您可以绑定公网的GitLab代码仓库，实现丰富的 Git 代码管理功能。'

- **访问凭证：**我们知道对于私有的 Docker Registry 有一个 docker login 命令需要输入用户名密码，而这个访问凭证就是用来设置 docker login 的密码相关的。可以设置固定密码和临时密码。固定密码相对来说更加方便，但是没有临时密码安全。大家可以根据实际情况使用。

The screenshot shows the 'Access Token' section of the AliCloud Docker Registry interface. The sidebar includes '容器镜像服务' (Container Image Service), '访问凭证' (Access Token) which is highlighted in blue, and '企业版实例' (Enterprise Edition Instances), '镜像中心' (Image Center), and '镜像搜索' (Image Search). The main content area has two numbered steps: 1. **获取凭证**: '在拉取私有镜像或者上传镜像前，需要docker login输入您的凭证信息。您可以选择通过 临时密码 或者 固定密码 作为访问凭证。' It includes sections for '固定密码' (Fixed Password) and '临时密码' (Temporary Password). 2. **登录实例**: '获取临时密码或固定密码后，可通过密码登录镜像服务实例：' with a note '1) 获取访问域名，基于当前的网络环境，选择对应的专有网络、公网或经典网络域名。' and '2) 在终端中输入访问凭证，登录Registry实例' followed by a command line example: '\$ sudo docker login --username=...@aliyun-test.com registry.cn-hangzhou.aliyuncs.com'.

基本设置

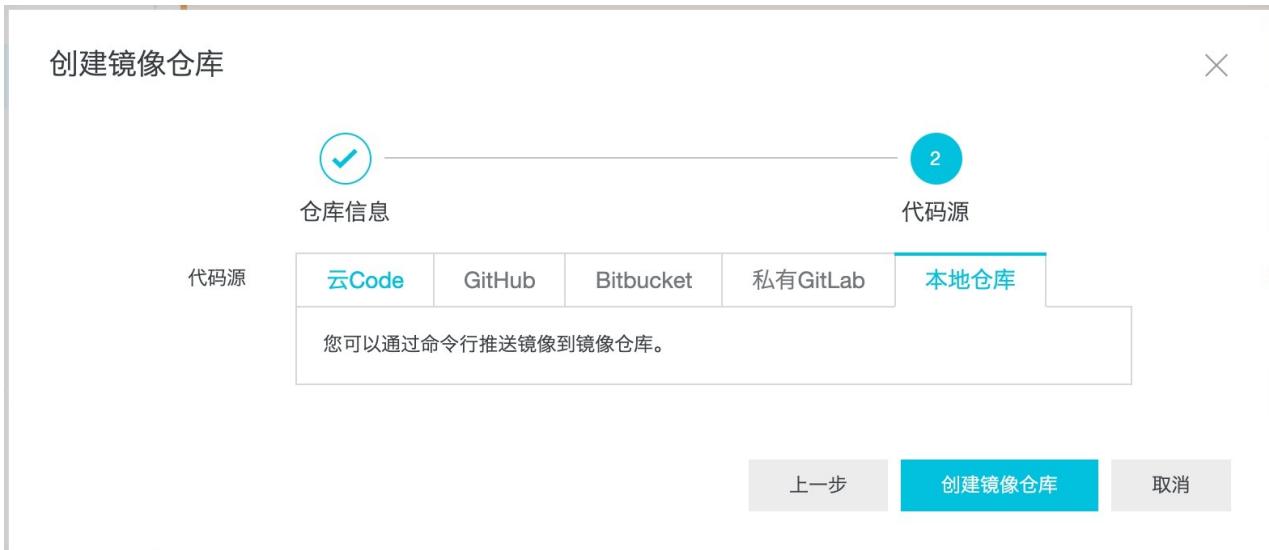
所以要真正使用这个 Docker Registry，我们需要先创建命名空间和镜像仓库。我们先创建一个命名空间叫 mooc-demo。

The screenshot shows the AliCloud Container Registry interface. On the left sidebar, under '默认实例' (Default Instance), the '命名空间' (Namespace) option is selected and highlighted in blue. The main content area is titled '命名空间' (Namespace) and lists two namespaces: 'mooc-demo' and another one whose name is partially visible. Both namespaces are in a 'Normal' state ('正常'). The 'mooc-demo' namespace has its '自动创建仓库' (Auto-create Repository) switch turned on ('开启') and is set to '私有' (Private). The second namespace also has its switch turned on and is set to '私有'. There are tabs for '镜像仓库' (Image Repository), '授权管理' (Authorization Management), and '代码源' (Code Source).

再创建一个镜像仓库叫 imooc-demo。

The screenshot shows the 'Create Image Repository' dialog. The left sidebar has '镜像仓库' (Image Repository) selected. The main form is titled '创建镜像仓库' (Create Image Repository). Step 1, '仓库信息' (Repository Information), is completed: the region is set to '华东1 (杭州)' (East China 1 Hangzhou) and the namespace is 'mooc-demo'. Step 2, '代码源' (Code Source), is partially visible. The repository name is 'imooc-demo'. A note indicates the name can be 2-64 characters long, using lowercase letters, numbers, underscores, hyphens, or periods. The repository type is set to '私有' (Private). The summary section contains the repository name 'imooc-demo'. The description section is empty and supports Markdown format.

下一步选择本地仓库。



创建完成之后我们可以看到镜像仓库的详细信息如下。我们可以看到镜像有多个地址：

- 公网地址：很容易理解，就是暴露在公网中。
- 专有网络：VPC 网络，VPC 可以简单理解一个特定的私有网络。
- 经典网络：经典网络是在 VPC 网络出来之前的内网，现在已经很少用了。

内网的好处是不需要上传下载镜像不需要走公网流量，毕竟网络流量收费还挺贵的。



imooc-demo

华东1（杭州） | 私有 | 本地仓库 | ● 正常

部署应用

基本信息

仓库授权

仓库名称 imooc-demo

公网地址

registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo

触发器

仓库地域 华东1（杭州）

专有网络

registry-vpc.cn-hangzhou.aliyuncs.com/mooc-demo/

镜像版本

仓库类型 私有

经典网络

registry-internal.cn-hangzhou.aliyuncs.com/mooc-demo/

镜像同步

代码仓库 无

摘要 imooc-demo

操作指南

镜像描述

1. 登录阿里云Docker Registry

```
$ sudo docker login --username=...@aliyun-test.com registry.cn-hangzhou.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-hangzhou.aliyuncs.com/mooc-demo/imooc-demo:[镜像版本号]
```

3. 将镜像推送到Registry

```
$ sudo docker login --username=ali-emr@aliyun-test.com registry.cn-hangzhou.aliyuncs.com
$ sudo docker tag [ImageId] registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo:[镜像版本号]
$ sudo docker push registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo:[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

从ECS推送镜像时，可以选择使用镜像仓库内网地址。推送速度将得到提升并且将不会损耗您的公网流量。

镜像上传下载

作为一个 Registry，它的最基本功能肯定是镜像的上传和下载，下面演示一下如何操作。

login

首先我们需要在本地先 login。

```
$ docker login --username=xxx registry.cn-hangzhou.aliyuncs.com
Password:
Login Succeeded
```

tag

要把本地的镜像上传到 Registry，需要将本地镜像重命名成远端镜像的形式，也就是上面截图中的第 3 步的操作。我们将本地的 busybox 镜像上传。

```
$ docker tag busybox registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo:v1
```

push

```
$ docker push registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo:v1
The push refers to repository [registry.cn-hangzhou.aliyuncs.com/mooc-demo/imooc-demo]
50761fe126b6: Pushed
v1: digest: sha256:2131f09e4044327fd101ca1fd4043e6f3ad921ae7ee901e9142e6e36b354a907 size: 527
```

这就表示上传上去了，我们从页面上面访问看一下。点击左侧的镜像版本可以看到最近的 v1 版本就是我们刚才上传的。

版本	镜像ID	状态	Digest	镜像大小	最后更新时间
v1	c7c37e472d31 ...	正常	2131f09e4044327fd101ca1fd4043e6f3ad921ae7ee901e9142e6e36b354a907	745.916 KB	2020-07-03 23:30:19

3. 总结

Docker 镜像仓库中心是我们使用 Docker 中不可或缺的一环。我们需要根据自身的情况选择合适的镜像中心。

比如我们涉及到容器的所有服务都在阿里云上面，那么镜像仓库中心肯定也使用阿里云的。

如果我们的容器应用在云上都属于 VPC 网络，那么我的镜像仓库中心也应该使用 VPC 网络。

}

06 萍花宝典：Docker 操作参数详解

08 站在巨人的肩膀上：热门 Docker 镜像介绍（一）