

[你好, 游客](#) [登录](#) [注册](#) [搜索](#)[首页](#) [Linux新闻](#) [Linux教程](#) [数据库技术](#) [Linux编程](#) [服务器应用](#) [Linux安全](#) [Linux下载](#) [Linux主题](#) [Linux壁纸](#) [Linux软件](#) [数](#)[首页](#) → [服务器应用](#)

阅读新闻

背景: ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

使用Docker Registry快速搭建私有镜像仓库

[日期: 2018-03-10]

来源: Linux社区 作者: ganbing

[字体: 大 中 小]

广告 X

彩丽琪 2017高跟鞋漆皮尖头高跟单鞋 细跟...
热销中
加入购物车

潮流奇遇 高跟鞋女2017 韩版甜美鞋细跟...
热销中
加入购物车

轻起跟鞋字扎
加

飞往欧洲所有主要城市

立即预订

谷歌广告

FIA

1、背景

在 Docker 中, 当我们执行 `docker pull xxx` 的时候, 可能会比较好奇, Docker 会去哪儿查找并下载镜像呢?

它实际上是从 `registry.hub.docker.com` 这个地址去查找, 这就是Docker公司为我们提供的公共仓库, 上面的镜像, 大家可以看到, 也可以使用。所以, 我们也可以带上仓库地址去拉取镜像, 如: `docker pull registry.hub.docker.com/library/alpine`, 不过要注意, 这种方式下载的镜像的默认名称就会长一些。

广告 X

[手机版](#) [繁體](#)

《Linux公社》

于默奥
价格从 5070 元起

爱丁堡
价格从 5230 元起

罗马
价格从 575 元起

点

书籍

卖

Python

实训班

教材

立即:

FINI

谷歌广告

加群加入: 811036767

最新资讯

Docker Swarm及界面管理工具Portainer安装
使用Docker Registry快速搭建私有镜像仓库
Docker轻量级图形页面管理Portainer安装配

如果要在公司中使用 **Docker**，我们基本不可能把商业项目上传到公共仓库中，那如果要多个机器共享，又能怎么办呢？

正因为这种需要，所以私有仓库也就有用武之地了。

所谓**私有仓库**，也就是在本地（局域网）搭建的一个类似公共仓库的东西，搭建好之后，我们可以将镜像提交到私有仓库中。这样我们既能使用 **Docker** 来运行我们的项目镜像，也避免了商业项目暴露出去的风险。

下面我们用官方提供的**registry**镜像来搭建私有镜像仓库，当然还有其它很多方法。

2、环境

准备两台安装好docker的服务器：

服务端机器（主机名为**registry**）：**docker**私有仓库服务器，运行**registry**容器；

测试端机器（主机名为**node**）：普通的**docker**服务器，在这台服务器上下载一个测试镜像**busybox**，然后上传到**registry**服务器进行测试；

3、部署(服务端操作)

3.1 下载镜像**registry**

```
[root@registry ~]# docker pull registry
Using default tag: latest
latest: Pulling from library/registry
81033e7c1d6a: Pull complete
b235084c2315: Pull complete
c692f3a6894b: Pull complete
ba2177f3a70e: Pull complete
a8d793620947: Pull complete
Digest: sha256:672d519d7fd7bbc7a448d17956ebee225d5eb27509d8dc5ce67ecb4a0bce54
Status: Downloaded newer image for registry:latest
```

3.2 查看镜下是否**pull**下来了

Ubuntu 16.04 下安装 Docker 17.12

Oracle 分析函数 ROW_NUMBER() 使用

MySQL数据库出现慢查询的危害

RHEL7.0 配置 ISCSI 详解

CentOS 7.4部署MariaDB Galera Cluster集群

C++STL之map型容器

Spring IOC容器的基本应用

```
[root@registry ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
php                  5.6.31             9ed5ccce1735       6 hours ago        1GB
nginx                1.12.1             19b0dc5eb6bb       7 hours ago        419MB
mysql                latest              5d4d51c57ea8       3 days ago         374MB
registry             latest              d1fd7d86a825       7 weeks ago        33.3MB
centos               7                  ff426288ea90       7 weeks ago        207MB
[root@registry ~]#
```

3.3 运行registry容器

```
[root@registry ~]# docker run -itd -v /data/registry:/var/lib/registry -p 5000:5000 --restart=always --name registry registry:latest
06a972de6218b1f1c3bf9b53eb9068dc66d147d14e18a89ab51db13e339d3dc9
```

参数说明

-itd: 在容器中打开一个伪终端进行交互操作，并在后台运行；

-v: 把宿主机的/data/registry目录绑定到容器/var/lib/registry目录(这个目录是registry容器中存放镜像文件的目录)，来实现数据的持久化；

-p: 映射端口；访问宿主机的5000端口就访问到registry容器的服务了；

--restart=always: 这是重启的策略，假如这个容器异常退出会自动重启容器；

--name registry: 创建容器命名为registry，你可以随便命名；

registry:latest: 这个是刚才pull下来的镜像；

3.4 测试镜像仓库中所有的镜像

```
[root@registry ~]# curl http://127.0.0.1:5000/v2/_catalog
{"repositories":[]}
```

现在是空的，因为才刚运行，里面没有任何镜像内容。

4、测试镜像仓库（测试端操作）

4.1 修改下镜像源并重启docker服务

```
[root@node ~]# vim /etc/docker/daemon.json
{
```

```
"registry-mirrors": [ "https://registry.docker-cn.com"]
}
```

```
[root@node ~]# systemctl restart docker
```

4.1 下载busybox镜像

```
[root@node ~]# docker pull busybox
```

```
[root@node ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
busybox	latest	f6e427c148a7	36 hours ago	1.15MB

4.2 为镜像打标签

```
[root@node ~]# docker tag busybox:latest 172.18.18.90:5000/busybox:v1
```

格式说明: Usage: docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

busybox:latest 这是源镜像, 也是刚才pull下来的镜像文件;

172.18.18.90:5000/busybox:v1: 这是目标镜像, 也是registry私有镜像服务器的IP地址和端口;

查看一下打好的tag:

```
[root@node ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
172.18.18.90:5000/busybox v1                 f6e427c148a7       36 hours ago       1.15MB
busybox              latest             f6e427c148a7       36 hours ago       1.15MB
[root@node ~]#
```

4.3 上传到镜像服务器

```
[root@node ~]# docker push 172.18.18.90:5000/busybox:v1
```

The push refers to repository [172.18.18.90:5000/busybox]

Get https://172.18.18.90:5000/v2/: http: server gave HTTP response to HTTPS client

注意了, 这是报错了, 需要https的方法才能上传, 我们可以修改下daemon.json来解决:

```
[root@node ~]# vim /etc/docker/daemon.json
```

```
{
  "registry-mirrors": [ "https://registry.docker-cn.com"],
```

```
"insecure-registries": [ "172.18.18.90:5000" ]
}
```

添加私有镜像服务器的地址，注意书写格式为json，有严格的书写要求，然后重启docker服务：

```
[root@node ~]# systemctl restart docker
```

在次上传可以看到没问题 了：

```
[root@node ~]# docker push 172.18.18.90:5000/busybox:v1
The push refers to repository [172.18.18.90:5000/busybox]
c5183829c43c: Pushed
v1: digest: sha256:c7b0a24019b0e6eda714ec0fa137ad42bc44a754d9cea17d14fba3a80ccc1ee4 size: 527
```

4.4 测试下载镜像

上传测试没问题了，我们接下来测试一下从registry服务器上下载刚才上传的busybox镜像，先删除node主机上的镜像：

```
[root@node ~]# docker rmi -f $(docker images -aq)
Untagged: 172.18.18.90:5000/busybox:v1
Untagged: 172.18.18.90:5000/busybox@sha256:c7b0a24019b0e6eda714ec0fa137ad42bc44a754d9cea17d14fba3a80ccc1ee4
Untagged: busybox:latest
Untagged: busybox@sha256:2107a35b58593c58ec5f4e8f2c4a70d195321078aebfadfbfb223a2ff4a4ed21
Deleted: sha256:f6e427c148a766d2d6c117d67359a0aa7d133b5bc05830a7ff6e8b64ff6b1d1d
Deleted: sha256:c5183829c43c4698634093dc38f9bee26d1b931dedeba71dbee984f42fe1270d
```

查看一下node主机上的镜像全部删除了：

```
[root@node ~]# docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
```

然后，从registry服务器上下载busybox镜像：

```
[root@node ~]# docker pull 172.18.18.90:5000/busybox:v1
v1: Pulling from busybox
d070b8ef96fc: Pull complete
Digest: sha256:c7b0a24019b0e6eda714ec0fa137ad42bc44a754d9cea17d14fba3a80ccc1ee4
```

Status: Downloaded newer image for 172.18.18.90:5000/busybox:v1

```
[root@node ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
172.18.18.90:5000/busybox	v1	f6e427c148a7	36 hours ago	1.15MB

列出所有镜像:

```
[root@node ~]# curl http://172.18.18.90:5000/v2/_catalog
```

```
{"repositories":["busybox"]}
```

列出**busybox**镜像有哪些**tag**:

```
[root@node ~]# curl http://172.18.18.90:5000/v2/busybox/tags/list
```

```
{"name":"busybox","tags":["v1"]}
```

更多**Docker**相关教程见以下内容:

Docker安装应用(CentOS 6.5_x64) <https://www.linuxidc.com/Linux/2014-07/104595.htm>

Ubuntu 16.04 服务器上配置使用 Docker <https://www.linuxidc.com/Linux/2017-06/145176.htm>

Ubuntu 15.04下安装Docker <https://www.linuxidc.com/Linux/2015-07/120444.htm>

Docker 安装实例 <https://www.linuxidc.com/Linux/2017-04/142666.htm>

Docker 创建基础镜像 <https://www.linuxidc.com/Linux/2017-05/144112.htm>

在 Ubuntu 15.04 上如何安装Docker及基本用法 <https://www.linuxidc.com/Linux/2015-09/122885.htm>

Ubuntu 16.04上Docker使用手记 <https://www.linuxidc.com/Linux/2016-12/138490.htm>

使用Docker分分钟启动常用应用 <https://www.linuxidc.com/Linux/2017-04/142649.htm>

Ubuntu 16.04下Docker修改配置文件不生效解决办法 <https://www.linuxidc.com/Linux/2017-05/143862.htm>

Docker 的详细介绍: [请点击这里](#)

Docker 的下载地址: [请点击这里](#)

本文永久更新链接地址: <https://www.linuxidc.com/Linux/2018-03/151308.htm>



**克拉科夫**
价格从 **4640 元起**

**马拉加**
价格从 **5430 元起**

500

谷歌广告



京东



绝对养眼Ubuntu美女壁纸



腾讯云云+校园扶持计划



Docker可视化管理工具
Portainer的安装配置及使用

广告 京东

linuxdc.com

广告 腾讯云

linuxdc.com



不止于OS X! 还有适用于Mac的八款替代操作系统



Linux下cmatrix的安装和使用



CentOS 7.4 发布下载，全稳定的Linux发行版_Linux下载



Ubuntu Mac (教程

linuxdc.com

linuxdc.com

linuxdc.com

linuxdc.com

关注Linux公社（LinuxIDC.com）官方微信与QQ群，随机发放邀请码

Docker轻量级图形页面管理Portainer安装配置

Docker Swarm及界面管理工具Portainer安装

相关资讯 Docker Registry Docker私有镜像仓库	
部署内网Docker Registry (08/09/2017 21:15:53)	搭建私有Docker Registry (07/27/2017 08:41:39)
企业级Docker Registry部署之 (06/20/2017 17:49:50)	局域网内部署 Docker Registry (05/23/2017 12:09:13)
搭建一个私有的Docker registry (05/04/2015 14:09:43)	如何部署 Docker Registry 服务 (04/28/2015 09:55:08)

本文评论 查看全部评论 (0)

表情： 姓名：  匿名 字数 0

https://www.linuxdc.com/Linux/2018-03/151308.htm

7/8

☒ 同意评论声明[请登录](#)

评论声明

- 尊重网上道德，遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任
- 本站管理人员有权保留或删除其管辖留言中的任意内容
- 本站有权在网站内转载或引用您的评论
- 参与本评论即表明您已经阅读并接受上述条款



[Linux公社简介](#) - [广告服务](#) - [网站地图](#) - [帮助信息](#) - [联系我们](#)

本站（LinuxIDC）所刊载文章不代表同意其说法或描述，仅为提供更多信息，也不构成任何建议。

主编：漏网的鱼 联系邮箱：root@linuxidc.net (如有合作请联系)

本站带宽由[\[808.Ai\]](#)友情提供

关注Linux，关注LinuxIDC.com，[请向您的QQ好友宣传LinuxIDC.com](#)，多谢支持！

Copyright © 2006-2017 Linux公社 All rights reserved 浙ICP备07014134号-8