



文章 (/blogs) > 大舒的博客 (/blog/qiukeke) > 文章详情

(/write)

Flux7 Docker 系列教程（四）：Registry & Workflows (/a/1190000002711379)



大舒 (/u/qiukeren) 3.2k 5月11日 发布

推荐

1 推荐

收藏

6 收藏, 432 浏览

本系列教程翻译自 Flux7 Docker Tutorial Series，系列共有九篇，本文译自第四篇 Part 4: Registry & Workflows (<http://blog.flux7.com/blogs/docker/docker-tutorial-series-part-4-registry-workflows>)。

该系列所有文章将参考其他学习资料翻译，也会加入自己的学习作为部分注解。如有错误，欢迎指正。



Docker Tutorials – Part 4

Registry & Workflows

上篇文章介绍了使用 Dockerfile 进行自动化部署和 Dockerfile 的常用命令，本篇文章来探讨 Docker Registry。类似 GitHub，Docker Registry (<http://Registry.hub.docker.com>) 是用来存储 Docker 镜像的地方。了解 Docker Registry 之前先来了解几个相关的知识吧。

1. 镜像和仓库和 GitHub 一样，可以被收藏和被“star”。
2. 和 GitHub 一样，可以在仓库上面留评论以便和维护人员进行交流。
3. 和 Github 类似私人仓库不能被搜索到，只有拥有合作者、拥有者权限的用户才能访问。
4. 推送成功之后可以配置 webhook (<http://en.wikipedia.org/wiki/Webhook>)。

Docker Registry 由三个部分组成：Index、Registry、Registry Client。

可以把 Index 认为是负责登录、负责认证、负责存储镜像信息和负责对外显示的外部实现，而 Registry 则是负责存储镜像的内部实现，而 Registry Client 则是 Docker 客户端。

Docker Hub 架构

Docker Index

Docker Index 使用以下工具来维护用户信息、校验镜像、以及维护公共 namespace：

1. 网页界面
2. 元数据存储
3. 认证服务
4. 令牌化

同时 Docker Index 也用来分解 URL，方便用户认证和用户使用。

Registry

Registry 用来存储镜像和统计数据。然而它不会提供数据库服务，即它不会提供用户认证服务，由 S3、云文件和本地文件系统提供数据库支持。另外，由 Index 通过 Token 进行用户认证。有多种不同的 Registry，下面是其中几个的例子：

1. **Sponsor Registry** 第三方 Registry，为客户和 Docker 社区提供服务。
2. **Mirror Registry** 第三方 Registry，只为客户提供服务。
3. **Vendor Registry** 由发布 Docker 镜像的供应商提供的 Registry。
4. **Private Registry** 通过多重安全检查进行认证的私有 Registry。

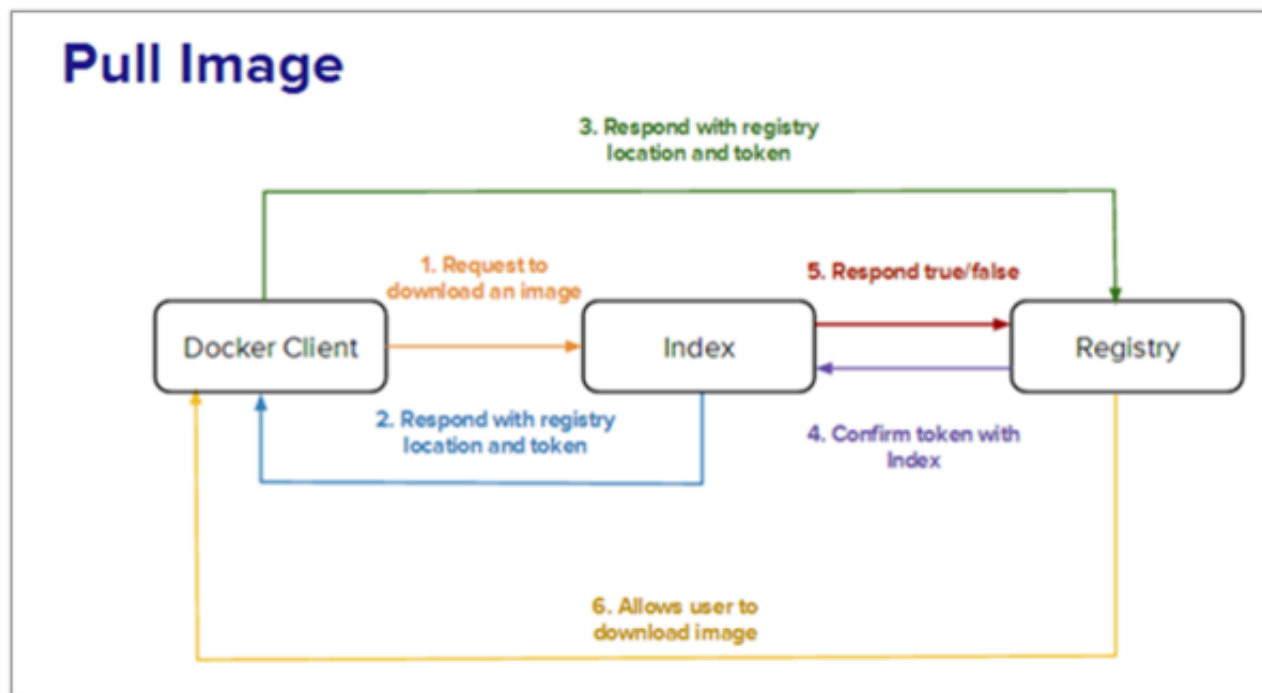
Registry Client

用于进行 pull 和 push 的操作，同时用于 docker push 时的登录认证。

为了更好地理解 Registry Client 的工作流程，下面将详细描述一下五个案例中的 Docker Registry 工作流程。

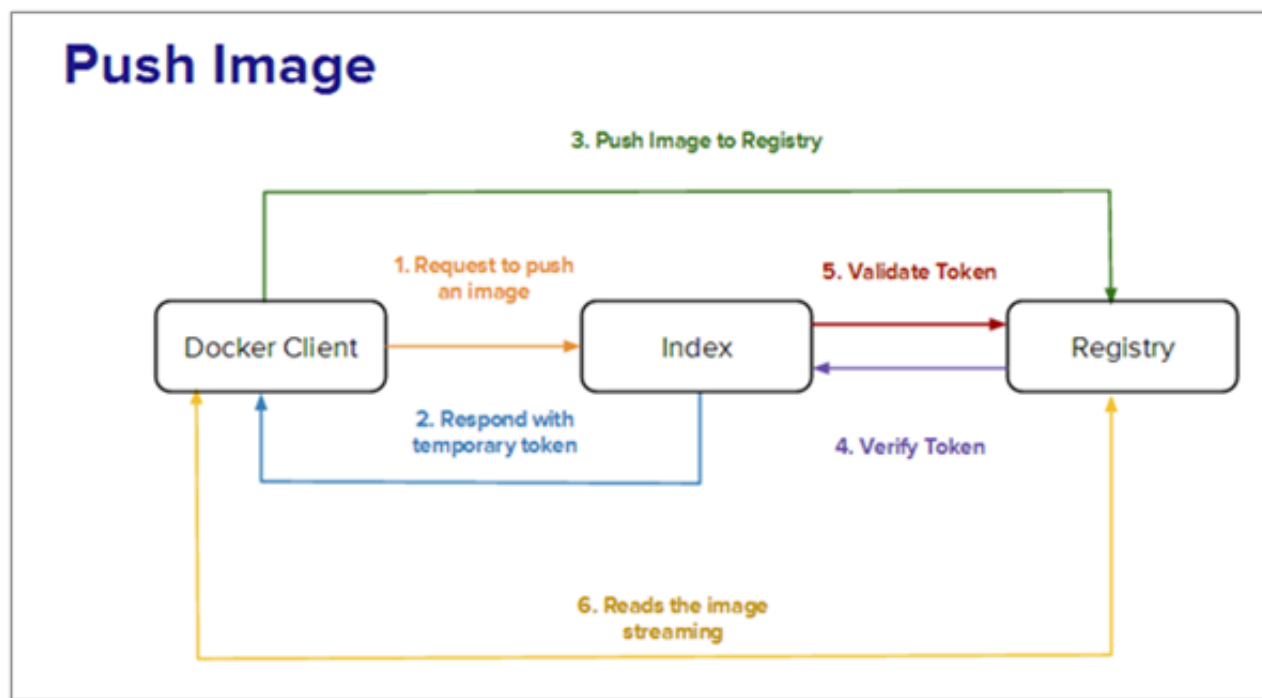
情景 A：从官方仓库 Pull 镜像

1. 用户对 Index 发出下载请求
2. Index 返回以下三个部分的信息：
 - 镜像所在的 Registry。
 - 镜像所有的层的校验
 - 认证之后的 Token
3. 用户使用 Token 和 Registry 进行通信，Registry 负责存储镜像和叠加在镜像上面的改动层。
4. Registry 确认是否是 Index 所发出的 Token。
5. Index 返回确认值，由此判断用户是否可以下载镜像。
>只有在请求的 header 里有 X-Docker-Token 时才会返回 Token。私有仓库需要认证，而公有仓库不需要认证。



情景 B：Push 镜像到 Registry

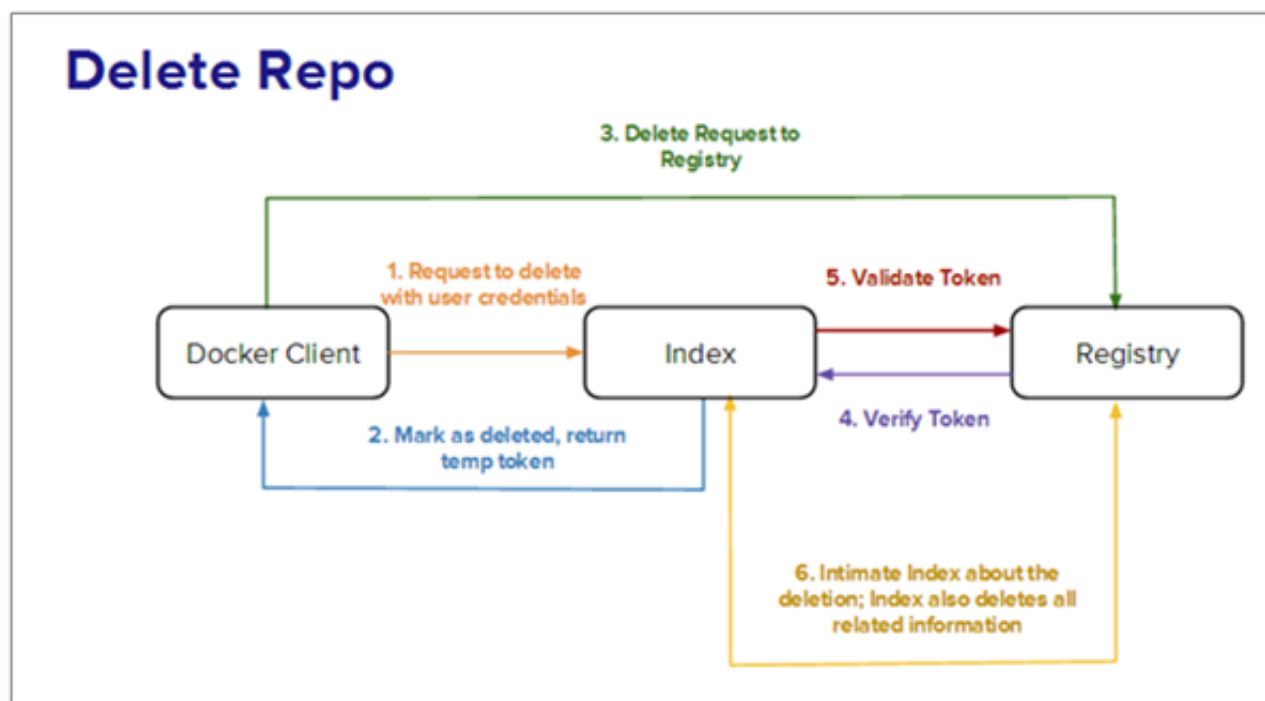
1. 用户向 Index 发送凭证，并要求分配 repository 名称。
2. 认证 namespace 的可用性后，分配 repository 名称，同时 Index 返回一个临时 Token。
3. 向 Registry 发送 Token 和镜像。
4. Registry 向 Index 确认 Token，确认完毕之后读取推送流。
5. Index 更新相关的镜像信息。



情景 C：从 Index 和 Registry 删除一个镜像

1. Index 收到 delete 某个 repository 的请求。
2. 进行 repository 认证和用户认证，成功后 Index 给客户端返回一个临时 Token。

3. Registry 收到删除信号和 Token。
4. Registry 向 Index 验证 Token，然后删除对应的镜像文件。
5. Registry 通知 Index 已经删除，Index 删除数据库中的所有相关的 repository 记录。



情景 D：在没有 Index 的情况下使用 Registry

没有 Index 的 Registry 是完全受 Docker Client 控制的。这种模式最适合在私有网络中存储镜像文件。Registry 运行在一个和 Index 没有通信的环境中，所有的认证问题和安全问题都需要用户自己解决。

情景 E：在有 Index 的独立模式中使用 Registry

在这种情况下，用户需要自己架设一个 Index 去解决存储和认证问题。当然，这种情况下可能会出现和官方 Index 时间不同步的问题。Docker 官方也提供了一个非常有意思的东西叫做 **chaining registries**，主要是为了解决负载均衡和为具体请求指定具体的 Registry。

当然，我们可以参考自己搭建本地 Docker Hub 服务 (<http://www.vpsee.com/2013/11/build-your-own-docker-private-registry-service/>) 来搭建本地的 Docker Hub。。

接下来的文章中将会继续介绍如何在以上每个场景中使用 Docker Registry API，并且也会深入了解 Docker 安全。

[docker \(/t/docker/blogs\)](#) [运维 \(/t/%E8%BF%90%E7%BB%B4/blogs\)](#) [linux \(/t/linux/blogs\)](#)

[链接 \(/a/1190000002711379\)](#) 更多 ▾

分享

1 推荐

收藏

本文由 [大舒 \(/u/qiukeren\)](#) 创作，采用 [知识共享署名-相同方式 3.0 中国大陆许可协议](#)

(<http://creativecommons.org/licenses/by-sa/3.0/cn>) 进行许可。
转载、引用前需联系作者，并署名作者且注明文章出处。

你可能感兴趣的文章

- Flux7 Docker系列教程（五）：Docker 安全 (/a/1190000002711383) 3 收藏，463 浏览
- Flux7 Docker 系列教程（一）：Docker 简介 (/a/1190000002711327) 21 收藏，1.4k 浏览
- docker中文文档++教程++中国镜像源++演示 (/a/1190000000404633) 13 收藏，7.1k 浏览

讨论区

添加评论

提交评论

语法提示

评论支持部分 Markdown 语法：
`**bold**` `_italic_` `[link](http://example.com)` `>` 引用 ``code``
- 列表。
同时，被你 @ 的用户也会收到通知



本文隶属于专栏
大舒的博客 (/blog/qiukeke)
我都不写PHP了你们还挤兑我。。

关注专栏

系列文章

Flux7 Docker 系列教程（八）：Docker Remote API (/a/1190000002711475) 3 收藏，399 浏览

Flux7 Docker 系列教程（七）：Docker API (/a/1190000002711455) 2 收藏，347 浏览

Copyright © 2011-2015 SegmentFault. 当前呈现版本 15.06.11

浙ICP备15005796号-2 (<http://www.miibeian.gov.cn/>)