

# Harbor - 企业级 Docker 私有仓库

## 一、安装底层需求

- Python应该是2.7或更高版本
- Docker引擎应为1.10或更高版本
- Docker Compose需要为1.6.0或更高版本

```
docker-compose: curl -L https://github.com/docker/compose/releases/download/1.9.0/docker-compose-`uname -s`-`uname -m`  
> /usr/local/bin/docker-compose
```

## 二、Harbor 安装：Harbor 官方地址：<https://github.com/vmware/harbor/releases>

### 1、解压软件包：tar xvf harbor-offline-installer-<version>.tgz

```
https://github.com/vmware/harbor/releases/download/v1.2.0/harbor-offline-installer-v1.2.0.tgz
```

### 2、配置harbor.cfg

#### a、必选参数

hostname: 目标的主机名或者完全限定域名

ui\_url\_protocol: http或https。默认为http

db\_password: 用于db\_auth的MySQL数据库的根密码。更改此密码进行任何生产用途

max\_job\_workers: (默认值为3) 作业服务中的复制工作人员的最大数量。对于每个映像复制作业，工作人员将存储库的所有标签同步到远程目标。增加此数字允许系统中更多的并发复制作业。但是，由于每个工作人员都会消耗一定数量的网络/CPU/IO资源，请根据主机的硬件资源，仔细选择该属性的值

customize\_cert: (on或off。默认为on) 当此属性打开时，prepare脚本将为注册表的令牌的生成/验证创建私钥和根证书

ssl\_cert: SSL证书的路径，仅当协议设置为https时才应用

ssl\_cert\_key: SSL密钥的路径，仅当协议设置为https时才应用

secretkey\_path: 用于在复制策略中加密或解密远程注册表的密码的密钥路径

### 3、创建 https 证书以及配置相关目录权限

```
openssl genrsa -des3 -out server.key 2048
```

```
openssl req -new -key server.key -out server.csr
```

```
cp server.key server.key.org
```

```
openssl rsa -in server.key.org -out server.key
```

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

```
mkdir /data/cert
```

```
chmod -R 777 /data/cert
```

### 4、运行脚本进行安装

```
./install.sh
```

### 5、访问测试

<https://reg.yourdomain.com> 的管理员门户 (将reg.yourdomain.com更改为您的主机名harbor.cfg)。请注意，默认管理员用户名/密码为admin / Harbor12345

### 6、上传镜像进行上传测试

a、指定镜像仓库地址

```
vim /etc/docker/daemon.json  
  
{  
  "insecure-registries": ["serverip"]  
}
```

b、下载测试镜像

```
docker pull hello-world
```

c、给镜像重新打标签

```
docker tag hello-world serverip/hello-world:latest
```

d、登录进行上传

```
docker login serverip
```

7、其它 Docker 客户端下载测试

a、指定镜像仓库地址

```
vim /etc/docker/daemon.json  
  
{  
  "insecure-registries": ["serverip"]  
}
```

b、下载测试镜像

```
docker pull serverip/hello-world:latest
```

### 三、Harbor 原理说明

#### 1、软件资源介绍

Harbor是VMware公司开源的企业级DockerRegistry项目，项目地址为<https://github.com/vmware/harbor>。其目标是帮助用户迅速搭建一个企业级的Dockerregistry服务。它以Docker公司开源的registry为基础，提供了管理UI，基于角色的访问控制(Role Based Access Control)，AD/LDAP集成、以及审计日志(Auditlogging)等企业用户需求的功能，同时还原生支持中文。Harbor的每个组件都是以Docker容器的形式构建的，使用Docker Compose来对它进行部署。用于部署Harbor的Docker Compose模板位于 /Deployer/docker-compose.yml，由5个容器组成，这几个容器通过Docker link的形式连接在一起，在容器之间通过容器名字互相访问。对终端用户而言，只需要暴露 proxy（即Nginx）的服务端口

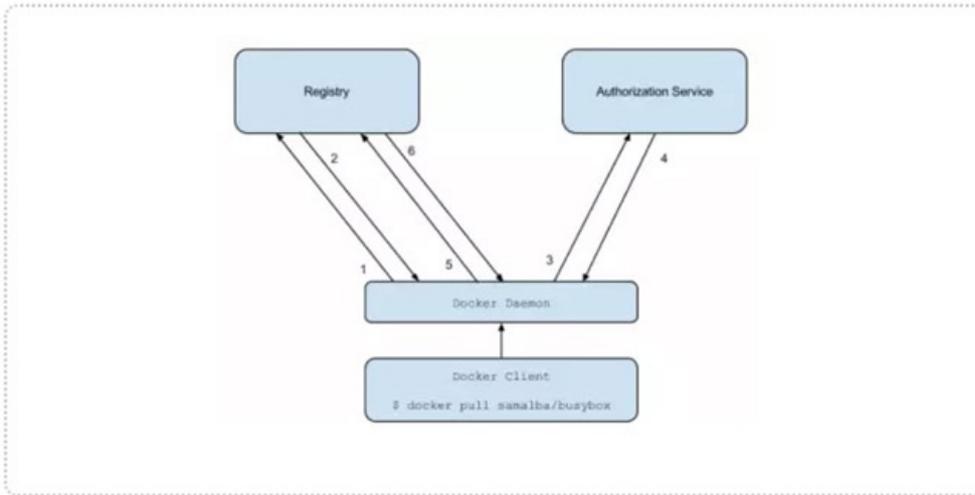
- Proxy: 由Nginx 服务器构成的反向代理。
- Registry: 由Docker官方的开源 registry 镜像构成的容器实例。
- UI: 即架构中的 core services，构成此容器的代码是 Harbor 项目的主体。
- MySQL: 由官方 MySQL 镜像构成的数据库容器。
- Log: 运行着 rsyslogd 的容器，通过 log-driver 的形式收集其他容器的日志

#### 2、Harbor 特性

- a、基于角色控制：用户和仓库都是基于项目进行组织的，而用户基于项目可以拥有不同的权限
- b、基于镜像的复制策略：镜像可以在多个Harbor实例之间进行复制
- c、支持LDAP：Harbor的用户授权可以使用已经存在LDAP用户
- d、镜像删除 & 垃圾回收：Image可以被删除并且回收Image占用的空间，绝大部分的用户操作API，方便

用户对系统进行扩展

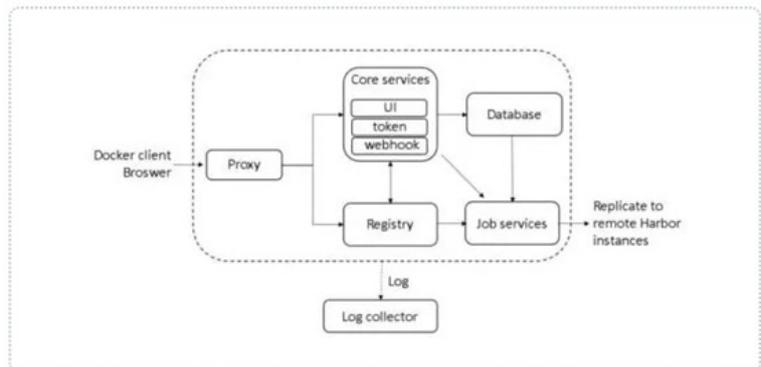
- e、用户UI: 用户可以轻松的浏览、搜索镜像仓库以及对项目进行管理
- f、轻松的部署功能: Harbor提供了online、offline安装, 除此之外还提供了virtualappliance安装
- g、Harbor和docker registry关系: Harbor实质上是对docker registry做了封装, 扩展了自己的业务模块



### 3、Harbor 认证过程

- a、dockerdaemon从docker registry拉取镜像。
- b、如果dockerregistry需要进行授权时, registry将会返回401 Unauthorized响应, 同时在响应中包含了docker client如何进行认证的信息。
- c、dockerclient根据registry返回的信息, 向auth server发送请求获取认证token。
- d、auth server则根据自己的业务实现去验证提交的用户信息是否存符合业务要求。
- e、用户数据仓库返回用户的相关信息。
- f、auth server将会根据查询的用户信息, 生成token令牌, 以及当前用户所具有的相关权限信息.上述就是完整的授权过程.当用户完成上述过程以后便可以执行相关的pull/push操作. 认证信息会每次都带在请求头中

### Harbor整体架构



### 4、Harbor 认证流程

- a、首先, 请求被代理容器监听拦截, 并跳转到指定的认证服务器。
- b、如果认证服务器配置了权限认证, 则会返回401。通知dockerclient在特定的请求中需要带上一个合法的token。而认证的逻辑地址则指向架构图中的core services。
- c、当docker client接受到错误code。client就会发送认证请求(带有用户名和密码)到coreservices进行basic auth认证。
- d、当C的请求发送给nginx以后, nginx会根据配置的认证地址将带有用户名和密码的请求发送到core services。
- e、coreservices获取用户名和密码以后对用户信息进行认证(自己的数据库或者介入LDAP都可以)。成功以后, 返回认证成功的信息

## Harbor认证流程

