## Docker镜像下载

我们使用registry：2.3进行仓库的搭建

$ sudo docker pull registry：2.3

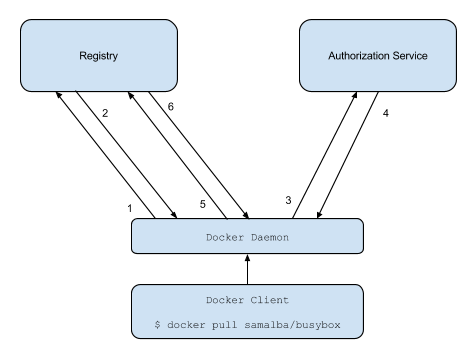
## Registry的部署

启动容器

$ docker run -d -v /opt/registry:/var/lib/registry -p 5000:5000 --restart=always --name registry registry:2.3

## docker registry v2 认证过程

目前docker registry v2 认证分为以下6个步骤：



1. docker client 尝试到registry中进行push/pull操作；

2. registry会返回401未认证信息给client(未认证的前提下)，同时返回的信息中还包含了到哪里去认证的信息；

3. client发送认证请求到认证服务器(authorization service)；

4. 认证服务器(authorization service)返回token；

5. client携带这附有token的请求，尝试请求registry；

6. registry接受了认证的token并且使得client继续操作；

### Step #1 ， Client 向registry 发起连接

通常，Docker Client在进行pull/push操作时，会先尝试连接docker registry。

Note: 当你访问远程的registry时，会用到tls验证域名的有效性(证书)，否则会出现如下错误:

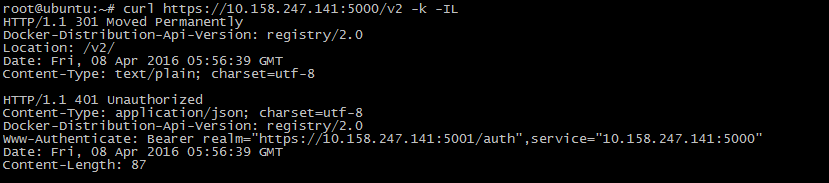
在docker的启动中加入下面的命令，来忽略对registry域名证书的审核:

--insecure-registry registry.example.com

### Step #2 ， 未认证响应(Unauthorized response)

Registry server会返回401并且会附带Authentication endpoint:

$ curl https://10.158.247.141:5000/v2 -k -IL （10.158.247.141:5000为registry地址）



Authentication server返回的头信息中包含了如何去或许token，它有几个重要的查询参数:

realm: #### 描述了认证的enpoint。

realm="https://10.158.247.141:5001/auth"

service: #### 描述了持有资源服务器的名称。

service="10.158.247.141:5000"

### Step #3&4 ， 认证endpoint通讯

需要明确的是，你需要知道：client发送请求到认证服务器签署token，请求信息中包含的基本身份验证信息将于服务器中的用户列表做匹配，然后根据请求中的scope要操作的范围、方法进而进行匹配，最后服务器匹配成功后将token进行签名，并且将token返回给客户端。

### Step #5&6 ， 最后沟通

Client尝试与registry连接(这次带了token信息)，registry接到请求后验证token，继而开始pull/push操作。

## 搭建带token认证的Registry

V2配置是通过配置文件和环境变量(docker run通过-e设置)来决定的

1. 修改配置文件方法：

docker run --name registry -v ./config.yml:/etc/docker/registry/config.yml registry:2

解释：把当前文件夹下的config.yml挂在到容器/etc/docker/registry/config.yml下，也就是替换了启动配置文件，也可以下载镜像源码在修改

1. 使用环境变量方法

在运行时使用-e REGISTRY\_variable设置环境变量，例如要设置：

storage:

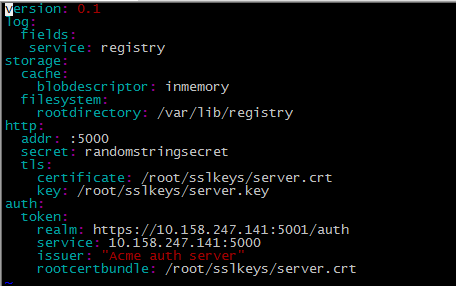
filesystem:

rootdirectory: /somewhere

对应：-e REGISTRY\_STORAGE\_FILESYSTEM\_ROOTDIRECTORY=/somewhere

我们采用配置文件的方式进行搭建：

1. 搭建registry



docker run -d -p 5000:5000 --restart=always

--name registry

-v /root/too/authserver:/root/sslkeys

-v /root/too/authserver/config.yml:/etc/docker/registry/config.yml

-v /opt/registry:/var/lib/registry registry:2.3

/root/too/authserver 为存放证书crt和key的路径

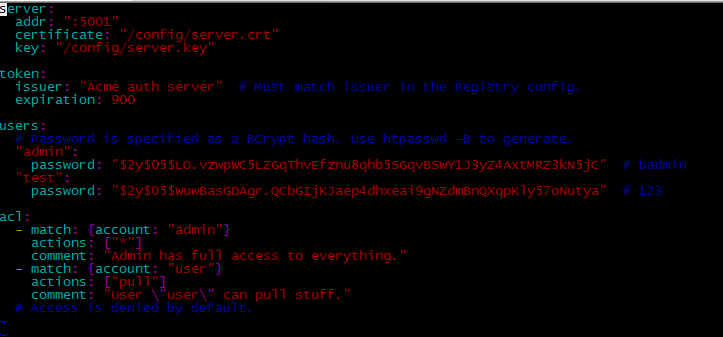
截图为config.yml的内容

realm表示我的auth server地址，service表示的registry的地址，issuer是一串标示符，随便写一下，auth server加密的时候也需要配置同样的字符串。rootcertbundle配置一个秘钥，对token进行加密。

1. 搭建auth-server

从github下找到了一个项目：<https://github.com/cesanta/docker_auth>

创建auth\_config.yml文件，内容如下



启动镜像：

docker run \

-it -d --name docker\_auth -p 5001:5001 \

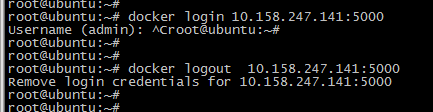
-v /root/too/authserver:/config:ro \

-v /var/log/docker\_auth:/logs \

cesanta/docker\_auth:stable /config/auth\_config.yml

这样带认证的registry搭建完成。

这样pull和push镜像需要进行docker login



当没有login，返回错误



Login后

