

1. 背景:

- 1.1 War包
- 1.2 Tomcat服务器
- 1.3 安装翻墙软件v2ray
- 1.4 在Ubuntu操作系统上安装Docker社区版

2. 实验:

- 2.1 打包制作docker镜像
- 2.2 docker镜像上传docker hub

1. 背景:

1.1 War包

War包一般是在进行Web开发时，通常是一个网站Project下的所有源码的集合，里面包含前台HTML/CSS/JS的代码，也包含Java的代码。

当开发人员在自己的开发机器上调试所有代码并通过后，为了交给测试人员测试和未来进行产品发布，都需要将开发人员的源码打包成War进行发布。

War包可以放在Tomcat下的webapps或者word目录下，随着tomcat服务器的启动，它可以自动被解压。简单理解就是一个web项目，其中包含web的所有东西。

1.2 Tomcat服务器

Tomcat服务器是一个免费的开放源代码的Web应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP程序的首选，最新的Servlet和JSP规范总能在Tomcat中得到体现。

1.3 安装翻墙软件v2ray

参考【Linux配置v2ray详细教程-Ubuntu为例】：<https://mahongfei.com/1776.html>

1. 下载QT实现的v2ray界面。在官网链接上找到最新的release下载。<https://github.com/Qv2ray/Qv2ray/releases>。比如，我目前下载的是Qv2ray.v2.2.3.linux-x64.AppImage。
2. 下载v2ray core核心文件。打开官方网站：<https://github.com/v2ray/v2ray-core/releases/>。选择下载文件v2ray-linux-64.zip。
3. 进入v2ray下载的根目录 `~/v2ray`，执行以下命令：

```
sudo chmod +x ./Qv2ray.v2.2.3.linux-x64.AppImage
```

4. 仍然在v2ray根目录下打开终端，输入以下命令：

```
./Qv2ray.v2.2.3.linux-x64.AppImage
```

5. 在Qv2ray的Preference页面的General Setting中，设置v2ray核心文件的路径。默认的路径是~/.config/qv2ray/vcore/。建议将v2ray-linux-64.zip解压后的文件都放置到这个目录里面。

```
cd ~/.config/  
mkdir qv2ray  
cd qv2ray  
mkdir vcore  
cd vcore  
mv path/to/v2ray-linux-64.zip .  
unzip v2ray-linux-64.zip
```

此时，用 `ls ~/.config/qv2ray/vcore/` 查验一下，目录下应该有可执行文件v2ray、v2ctl、v2ray.sig、v2ctl.sig。最后，在Qv2ray的设置窗口中点击Check V2ray Core Setting按钮，验证这两个可执行文件的有效性。

6. 打开Qv2ray的主界面。顶部有个subscriptions按钮。在弹出的SubscribeEditor窗口中新建一个subscription。修改Name为zion ladder，修改Address为梯子的链接字符串。然后点击Update Subscription Data，获得更新的Connection List。之后点击OK，关闭SubscribeEditor窗口。
7. 在Qv2ray主窗口中，选择默认连接，比如香港阿里云-001，连接比较顺畅。
8. 在Qv2ray的Preferences窗口中，找到Inbound Setting的Tab。勾选Set System Proxy选项，打开系统全局的网络翻墙功能。此时可以在Chrome浏览器上，打开google.com或者youtube.com，测试是否翻墙成功。
9. 如果想仅仅在Chrome浏览器上做网络翻墙功能，就在Chrome浏览器上安装Proxy SwitchyOmega插件。在插件的配置页面上，设置Proxy Server，采用协议SOCK5，代理服务器是本地127.0.0.1，端口是1089（具体端口号查看Qv2ray的Preferences窗口的Inbound Setting的Tab，看SOCKS Settings）。浏览器代理设置完成以后，如果不想做全局的System Proxy，就在Qv2ray的Preferences窗口，去掉Set System Proxy选项。

1.4 在Ubuntu操作系统上安装Docker社区版

参考：<https://docs.docker.com/engine/install/ubuntu/>

其他操作系统上的Docker安装，要自己另外查步骤。

前期准备

```
sudo apt-get update  
  
sudo apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    curl \  
    gnupg-agent \  
    gnupg2 pass\  
    software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

sudo add-apt-repository \
    "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) \
    stable"
```

安装docker engine (community edition) 免费的社区版

```
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

下面的命令可以列举所有不同版本docker engine (community edition)的引擎

```
$ apt-cache madison docker-ce

docker-ce | 5:20.10.4~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:20.10.3~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:20.10.2~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:20.10.1~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:20.10.0~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:19.03.15~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:19.03.14~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 5:19.03.13~3-0~ubuntu-bionic |
https://download.docker.com/linux/ubuntu bionic/stable amd64 Packages
```

也可以不选择最新版本的docker engine，转而选择特定版本的docker-ce引擎安装，比如选择 `docker-ce=5:20.10.4~3-0~ubuntu-bionic`

```
sudo apt-get install docker-ce=<VERSION_STRING> docker-ce-cli=
<VERSION_STRING> containerd.io
```

去docker hub申请一个免费账号。<https://hub.docker.com/> 再用下面的命令，在本地docker登录账号。

```
#Open up shell in administrator mode
sudo -s
docker login
# Enter Docker ID & Password.
exit
```

如果docker login无法访问docker hub网站。一般是命令行下面没有梯子工具。一般需要设置http_proxy和https_proxy。

走 http代理，具体查看自己的梯子工具的设置页面。下面假设了翻墙工具的http代理的端口是8888。自己要根据实际情况修改端口号。

```
git config --global http.proxy "http://127.0.0.1:8888"
git config --global https.proxy "http://127.0.0.1:8888"

export http_proxy="http://127.0.0.1:8888"
export https_proxy="http://127.0.0.1:8888"
```

或者走 socks5 代理（如 Shadowsocks）。下面假设了翻墙工具的sock5代理的端口是1089。自己要根据实际情况修改端口号。

```
git config --global http.proxy 'socks://127.0.0.1:1089'
git config --global https.proxy 'socks://127.0.0.1:1089'

export socks5="socks5://127.0.0.1:1089"
```

It is also OK to use proxy url `socks5://127.0.0.1:1089`. [According to this document](#), despite the name `http.proxy`, it should work for both HTTP and HTTPS repository urls.

测试docker run。

```
sudo docker info
sudo docker run hello-world
```

要熟悉docker可以先试试下面的[docker101tutorial](#)里面的步骤。

```
git clone https://github.com/docker/getting-started
cd getting-started
docker build -t docker101tutorial .
docker run -d -p 80:80 --name docker-tutorial docker101tutorial
```

2. 实验:

2.1 打包制作docker镜像

- 创建一个 Dockerfile 文件

```
# 先创建一个文件夹为docker-admin
mkdir docker-admin

# 进入文件夹docker-admin 并创建一个Dockerfile
cd docker-admin && vim Dockerfile
```

vim 指令:

:w 保存文件但不退出vi :w file 将修改另外保存到file中, 不退出vi :w! 强制保存, 不推出vi
:wq 保存文件并退出vi :wq! 强制保存文件, 并退出vi :q 不保存文件, 退出vi :q! 不保存文件, 强制退出vi :e! 放弃所有修改, 从上次保存文件开始再编辑命令历史

- 在新建的 Dockerfile 文件里, 插入以下命令。

```
FROM docker.io/tomcat
MAINTAINER rstyro:
COPY admin.war /usr/local/tomcat/webapps
```

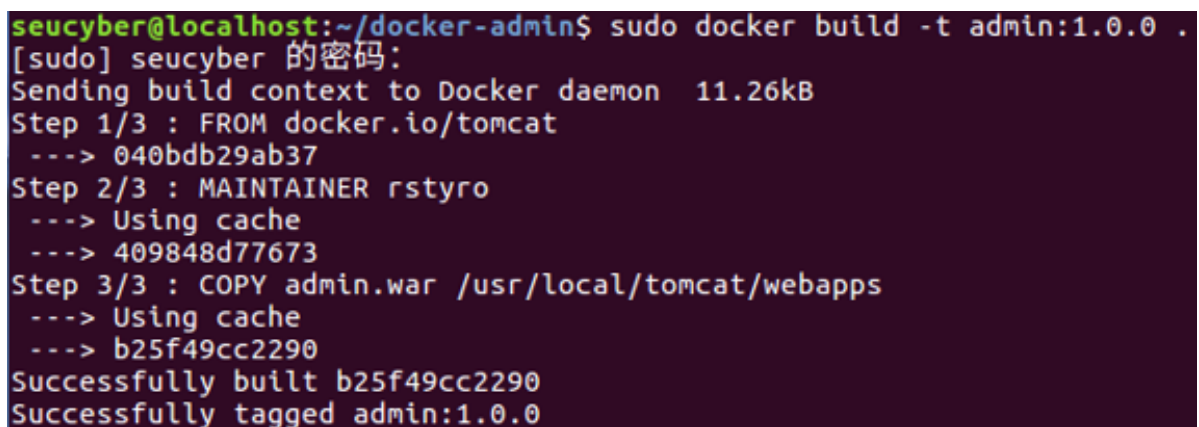
- 获取到 .war 文件

```
mv xxx.war admin.war
```

- 构建和运行镜像

```
# -t 参数 后面跟镜像名字和tag 注意别忘了后面的 . 点表示当前路径
docker build -t admin:1.0.0 .
# 镜像取名 admin 本机端口映射 8080
docker run --name=admin -p 8080:8080 -d admin:1.0.0
```

本地构建镜像成功:



```
seucyber@localhost:~/docker-admin$ sudo docker build -t admin:1.0.0 .
[sudo] seucyber 的密码:
Sending build context to Docker daemon 11.26kB
Step 1/3 : FROM docker.io/tomcat
--> 040bdb29ab37
Step 2/3 : MAINTAINER rstyro
--> Using cache
--> 409848d77673
Step 3/3 : COPY admin.war /usr/local/tomcat/webapps
--> Using cache
--> b25f49cc2290
Successfully built b25f49cc2290
Successfully tagged admin:1.0.0
```

sudo docker image ls 指令可以查看本地的docker镜像。

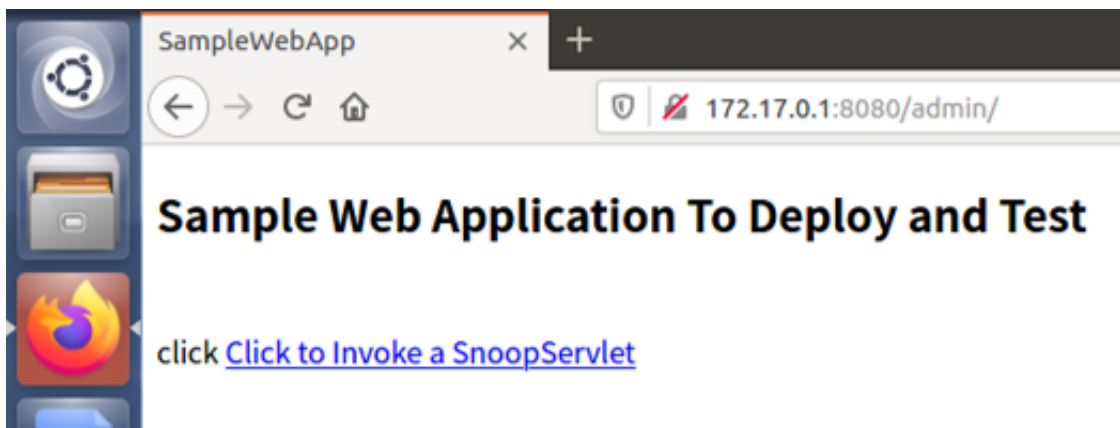
```
seucyber@master:~/docker-admin$ sudo docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
cscyp/admin1.0.0    latest             b25f49cc2290       About an hour ago   649MB
admin                1.0.0             b25f49cc2290       About an hour ago   649MB
tomcat               latest             040bdb29ab37       8 weeks ago        649MB
bestwu/wechat        latest             53c371b7016c       18 months ago      941MB
bestwu/qq            office             dia0bdfead00       2 years ago        792MB
```

此时已成功将 war 包部署到Tomcat，可通过 #ifconfig 查看网卡 ip 后在本地浏览器 localhost 查看。

```
seucyber@localhost:~/docker-admin$ ifconfig
docker0  Link encap:以太网 硬件地址 02:42:cc:34:d7:27
          inet 地址:172.17.0.1 广播:172.17.255.255 掩码:255.255.0.0
          inet6 地址: fe80::42:ccff:fe34:d727/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:75211 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:136456 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:0
          接收字节:3593164 (3.5 MB) 发送字节:205310462 (205.3 MB)
```

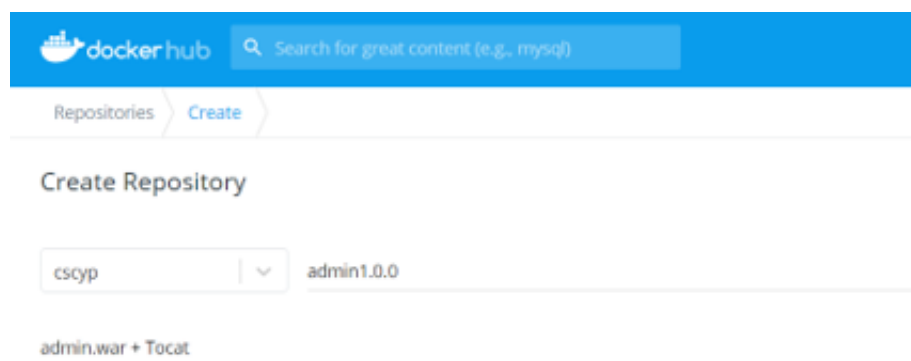
例如：我本地的网卡ip：172.17.0.1，在浏览器访问 172.17.0.1:8080/admin/

访问成功界面如下：



2.2 docker镜像上传docker hub

- 首先申请一个docker hub 帐号，登录到dockehub点击一下按钮：create —> create repository，取个名字，我起名为 admin1.0.0，可以选择增加对镜像的描述。



- 镜像打标签

#这里的tag不指定默认为latest

```
docker tag <existing-image> <hub-user>/<repo-name>[:<tag>]
```

- push镜像

1、本地登录 docker hub 账号，命令如下：

```
root@master:~# docker login
Username: 账号名
Password: 密码
Email: 邮箱地址
WARNING: login credentials saved in /root/.docker/config.json
Login Succeeded
```

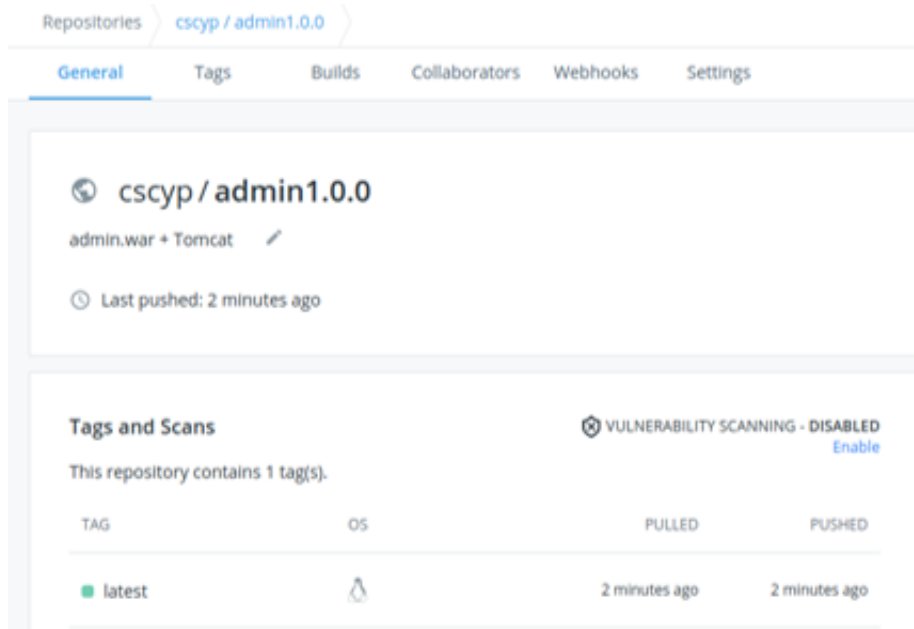
2、推送镜像，命令如下：

```
docker push <hub-user>/<repo-name>:<tag>
```

```
seucyber@master:~/docker-admin$ sudo docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, he
ad over to https://hub.docker.com to create one.
Username: cscyp
Password:
WARNING! Your password will be stored unencrypted in /home/seucyber/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
seucyber@master:~/docker-admin$ sudo docker push cscyp/admin1.0.0
The push refers to repository [docker.io/cscyp/admin1.0.0]
1b58ba180c4e: Pushed
9ddc8cd8299b: Mounted from library/tomcat
c9132b9a3fc8: Mounted from library/tomcat
8e2e6f2527c7: Mounted from library/tomcat
500f722b156b: Mounted from library/tomcat
7a9b35031285: Mounted from library/tomcat
7496c5e8691b: Mounted from library/tomcat
aa7af8a465c6: Mounted from library/tomcat
ef9a7b8862f4: Mounted from library/tomcat
a1f2f42922b1: Mounted from library/tomcat
4762552ad7d8: Mounted from library/tomcat
latest: digest: sha256:632714d281ba3b34ec8ca1648a25e401adc98c18c5c3e6d0f44a9e84ca85cc12 size: 2629
```

- 登录验证或下拉验证：



或下拉镜像也可验证：

```
docker pull <hub-user>/<repo-name>:<tag>
```