Continuous Monitoring Using Nagios

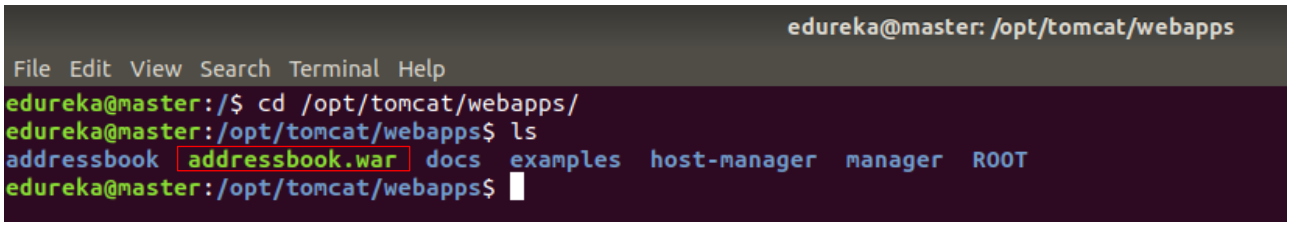
Demo Document 2

edureka!

Demo -2: Monitor your webapp using check_http and check_tcp plugins of Nagios

Step 1: Check for the addressbook.war file in your tomcat app within webapps folder.

```
cd /opt/tomcat/webapps/
ls
```



Step 2: Copy your addressbook.war to a location where you want to create a dockerfile.

```
sudo mkdir addressbook
cd addressbook
sudo cp /opt/tomcat/webapps/addressbook.war
```

Step 3 : Create a dockerfile using below commands:

```
sudo touch Dockerfile
```

Step 4 : Now edit your dockerfile and write following commands inside to create addressbook application image.

```
sudo nano Dockerfile

FROM library/tomcat
ADD . /usr/local/tomcat/webapps/
CMD "Catalina.sh" "run"
```

```
edureka@master: /addressbook
File  Edit  View  Search  Terminal  Help
  GNU nano 2.8.6                                File: Dockerfile

FROM library/tomcat
ADD  . /usr/local/tomcat/webapps/
CMD "catalina.sh" "run"
```

Step 5: Save your Dockerfile and build it to create an image by using the below command.

```
docker build -t edureka/addressbook .
```

```
edureka@master: /addressbook
File  Edit  View  Search  Terminal  Help
edureka@master:/addressbook$ docker build -t edureka/addressbook .
Sending build context to Docker daemon  16.54MB
Step 1/3 : FROM library/tomcat
 ---> 78b258e36eed
Step 2/3 : ADD  . /usr/local/tomcat/webapps/
 ---> Using cache
 ---> 208cd7104ca8
Step 3/3 : CMD "catalina.sh" "run"
 ---> Using cache
 ---> 5985bb3cd54d
Successfully built 5985bb3cd54d
Successfully tagged edureka/addressbook:latest
edureka@master:/addressbook$
```

Step 6: Check your Image by using the below command.

```
docker images
```

```
edureka@master: /addressbook
File  Edit  View  Search  Terminal  Help
edureka@master:/addressbook$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
edureka/addressbook     latest              5985bb3cd54d        45 minutes ago      479MB
```

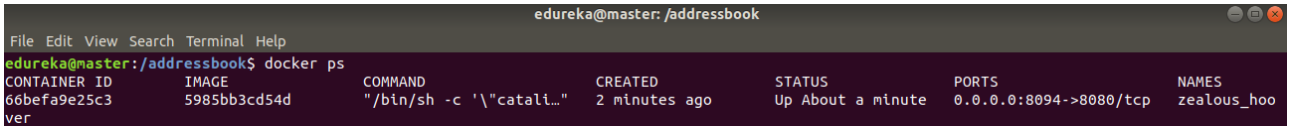
Step 7: Create a container from the image using below commands.

```
docker run -it -d -p 8094:8080
```

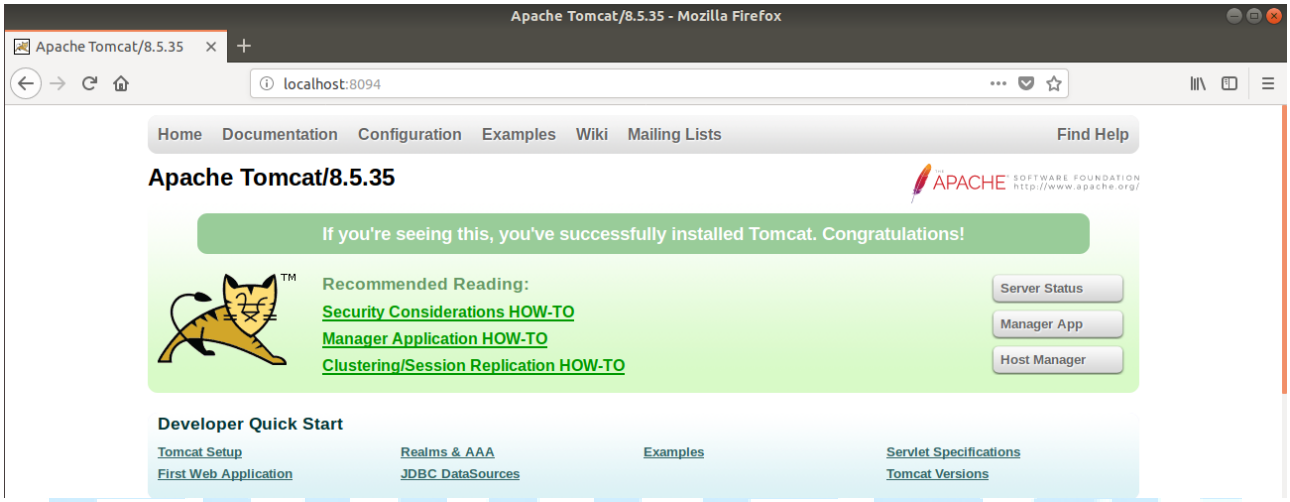
```
edureka@master: /addressbook
File  Edit  View  Search  Terminal  Help
edureka@master:/addressbook$ docker run -it -d -p 8094:8080 5985bb3cd54d
66befa9e25c3151b2e9a6c997efa2a8a29516276e18b83e9639e69c8110a6b1b
edureka@master:/addressbook$
```

Step 8: Check your container by using below commands:

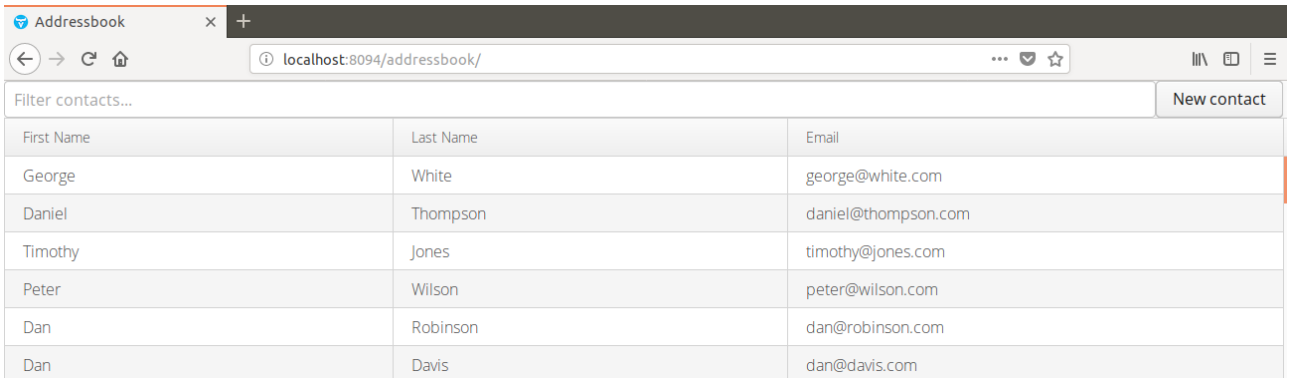
```
docker ps
```



Step 9: Now go to the ip address 8094 on your browser.

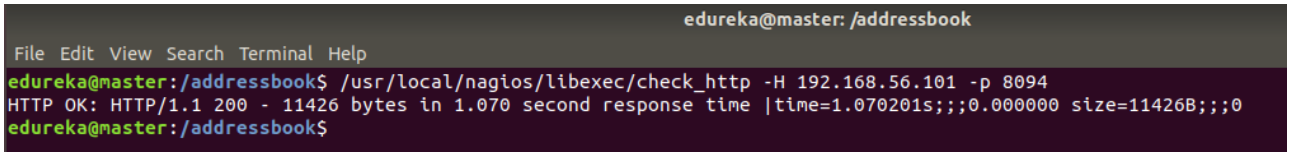


Step 10: Run that addressbook application by typing `localhost:8094/addressbook` on your browser. You will get screen like below.



Step 11: Now go back to your terminal and type following command to check if your localhost is able to connect to the running tomcat container on the browser.

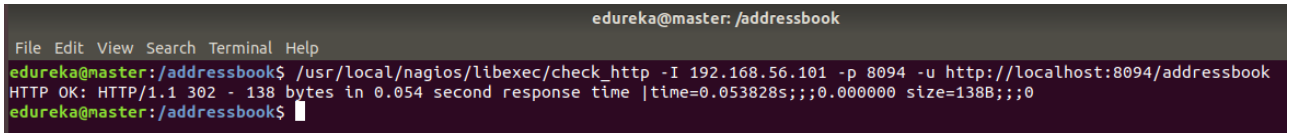
```
/usr/local/nagios/libexec/check_http -H 192.168.56.101 -p 8094
```



Since -H refers to the hostname here and -p is referring to the port which we want to check the connection with.

Step 12: Check the url of your application if it's working fine or not using the below commands.

```
/usr/local/nagios/libexec/check_http -I 192.168.56.101 -p 8094 -u
http://localhost:8094/addressbook
```

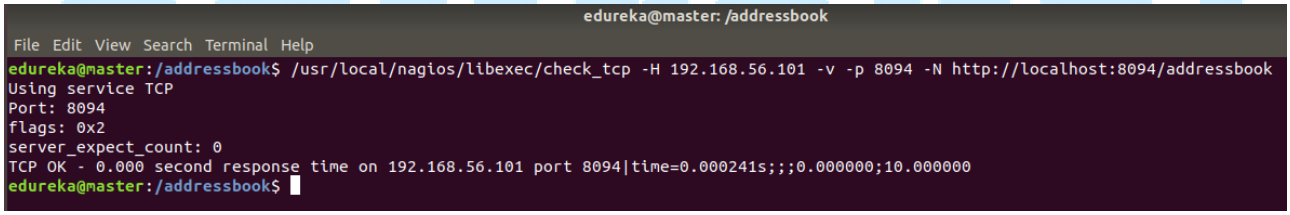


In the above screenshot -u has been used to specify the url of the application.

You can check for tcp connections as well using the same command as check_http. Just replace http with tcp and you are good to go.

Step 13: Use the below command to check the tcp status of your application:

```
/usr/local/nagios/libexec/check_tcp -H 192.168.56.101 -v -p 8094 -N
http://localhost:8094/addressbook
```



You can check tcp connection for your tomcat server in the same way as above but since you are able to access your application which means your tomcat server is obviously running.