Creating Docker Container for ACL Tool

This document would guide you to setup docker on your windows, linux or Mac OS, with our ACL Tool and its dependencies.

We are going to use standard **Ubuntu:20.04**

Creating a Dockerfile – For all platforms

Open a notepad or any text editor, please put the below contents in the file and save the file as Dockerfile, without any extension.

– –

```
FROM ubuntu:20.04

MAINTAINER Siddharth Joshi

ARG DEBIAN_FRONTEND=noninteractive

ENV TZ=Europe/Dublin

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

RUN apt update && \
    apt install -y net-tools mtr htop strace && \
    apt install -y python3 && \
    apt install -y vim python3-pip && \
    apt-get install -y build-essential libssl-dev libffi-dev software-properties-common git && apt-get clean

WORKDIR /opt/

RUN git clone https://github.com/1982League/FYP-2022.git

RUN git clone https://github.com/networktocode/ntc-templates.git

RUN git clone https://github.com/google/capirca.git

WORKDIR /opt/FYP-2022/

RUN chmod +x *.py

RUN chmod +x *.sh

RUN ./setup.sh
```

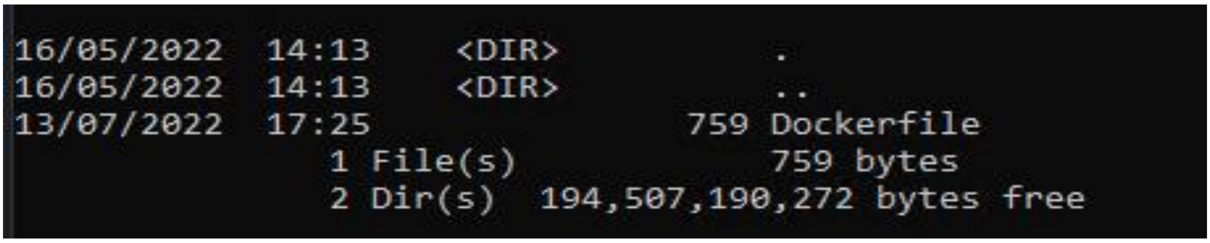
– –

Open a Command Prompt from the windows, go to the same directory as the ***Dockerfile*** file is created on.

Click on start button and, type ***cmd*** (Command Prompt).

Once you are on the command prompt, make sure the Dockerfile is there, by checking with ***dir*** command.

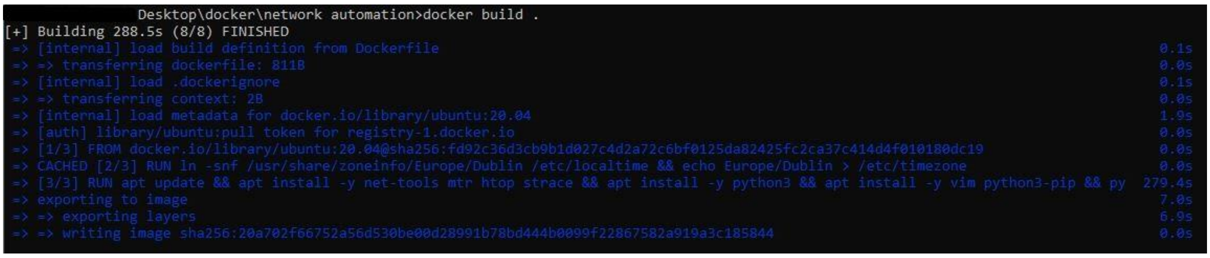
Note: We are creating DOCKERFILE on windows machine, the process is the same for all OS.



```
16/05/2022  14:13    <DIR>          .
16/05/2022  14:13    <DIR>          ..
13/07/2022  17:25               759 Dockerfile
               1 File(s)            759 bytes
               2 Dir(s)  194,507,190,272 bytes free
```

Figure 1 DOCKERFILE

Once confirmed the file is there!! type ***docker build .***
on the command prompt, (.) denotes the docker build will be in the current directory.



```
[redacted] Desktop\docker\network automation>docker build .
[+] Building 288.5s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                                   0.1s
 => => transferring dockerfile: 811B                                                                                   0.0s
 => [internal] load .dockerignore                                                                                      0.1s
 => => transferring context: 2B                                                                                        0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                        1.9s
 => [auth] library/ubuntu:pull token for registry-1.docker.io                                                          0.0s
 => [1/3] FROM docker.io/library/ubuntu:20.04@sha256:fd92c36d3cb9b1d027c4d2a72c6bf0125da82425fc2ca37c414d4f010180dc19  0.0s
 => CACHED [2/3] RUN ln -snf /usr/share/zoneinfo/Europe/Dublin /etc/localtime && echo Europe/Dublin > /etc/timezone    0.0s
 => [3/3] RUN apt update && apt install -y net-tools mtr htop strace && apt install -y python3 && apt install -y vim python3-pip && py  279.4s
 => exporting to image                                                                                                 7.0s
 => => exporting layers                                                                                                6.9s
 => => writing image sha256:20a702f66752a56d530be00d28991b78bd444b0099f22867582a919a3c185844                          0.0s
```

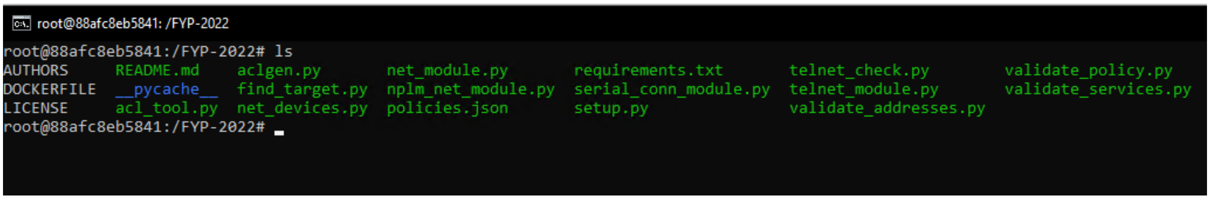
Figure 2 Successful Docker Build

Docker Image: ***20a702f66752a56d530be00d28991b78bd444b0099f22867582a919a3c185844***

To list out the docker process – ***docker ps -a***

C:\docker\network automation>docker run -it 20a702f66752a56d530be00d28991b78bd444b0099f22867582a919a3c185844 bash

root@88afc8eb5841:/#



```
root@88afc8eb5841: /FYP-2022
root@88afc8eb5841:/FYP-2022# ls
AUTHORS     README.md    aclgen.py       net_module.py        requirements.txt      telnet_check.py         validate_policy.py
DOCKERFILE  __pycache__  find_target.py  nplm_net_module.py   serial_conn_module.py telnet_module.py        validate_services.py
LICENSE     acl_tool.py  net_devices.py  policies.json        setup.py              validate_addresses.py
root@88afc8eb5841:/FYP-2022# _
```

Figure 3 ACL Tool Docker Container

This container can be setup on Linux, Windows, or MAC OS.

ACL Tool Setup on Docker – Linux

After creating a Dockerfile on linux.

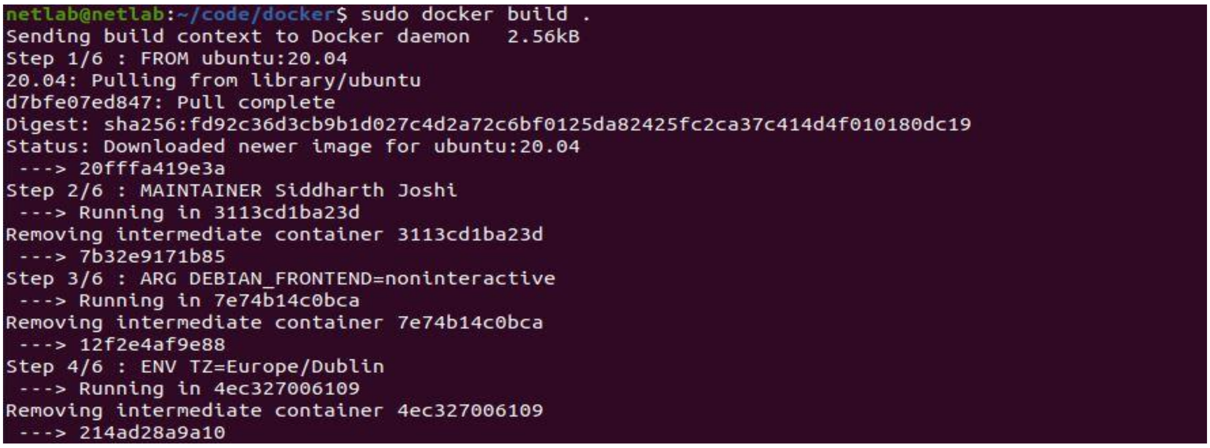
Make sure you are in the same directory as the Dockerfile and type in the command given below.

docker build .

Or

sudo docker build .

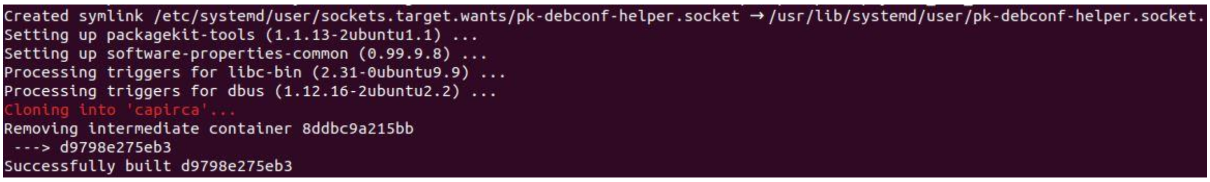
(.) denotes the docker build will be in the current directory in our case /home/NetworkAutomation/. '#' you don't need to type hash symbol as, it means you are in elevated privileged and logged in as a root user.



```
netlab@netlab:~/code/docker$ sudo docker build .
Sending build context to Docker daemon   2.56kB
Step 1/6 : FROM ubuntu:20.04
20.04: Pulling from library/ubuntu
d7bfe07ed847: Pull complete
Digest: sha256:fd92c36d3cb9b1d027c4d2a72c6bf0125da82425fc2ca37c414d4f010180dc19
Status: Downloaded newer image for ubuntu:20.04
 ---> 20fffa419e3a
Step 2/6 : MAINTAINER Siddharth Joshi
 ---> Running in 3113cd1ba23d
Removing intermediate container 3113cd1ba23d
 ---> 7b32e9171b85
Step 3/6 : ARG DEBIAN_FRONTEND=noninteractive
 ---> Running in 7e74b14c0bca
Removing intermediate container 7e74b14c0bca
 ---> 12f2e4af9e88
Step 4/6 : ENV TZ=Europe/Dublin
 ---> Running in 4ec327006109
Removing intermediate container 4ec327006109
 ---> 214ad28a9a10
```

Figure 4 Docker build on Linux - Ubuntu

The docker build may take few mins, Wait to see the message shown below.



```
Created symlink /etc/systemd/user/sockets.target.wants/pk-debconf-helper.socket → /usr/lib/systemd/user/pk-debconf-helper.socket.
Setting up packagekit-tools (1.1.13-2ubuntu1.1) ...
Setting up software-properties-common (0.99.9.8) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for dbus (1.12.16-2ubuntu2.2) ...
Cloning into 'capirca'...
Removing intermediate container 8ddbc9a215bb
 ---> d9798e275eb3
Successfully built d9798e275eb3
```

Figure 5 Docker Image created

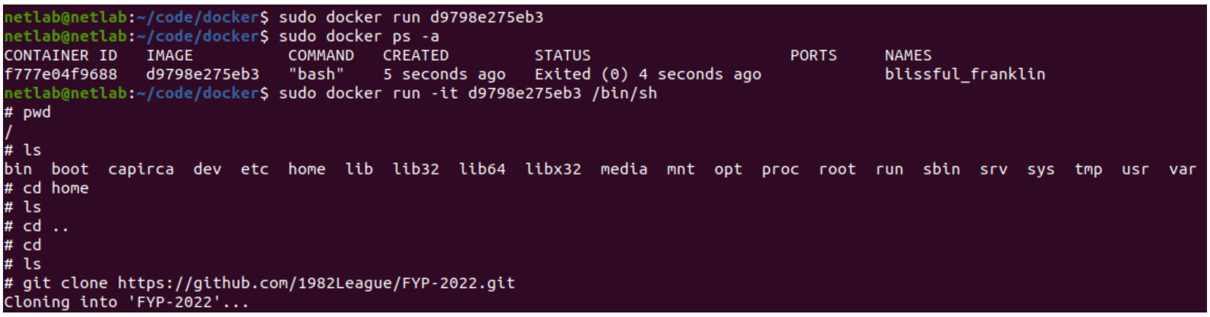
Docker Image: **d9798e275eb3 created successfully.**

Note: There is a bit of change with docker file naming, DOCKERFILE in windows works, I was having issue and I had to change the filename in the linux to Dockerfile.

To run the docker container in interactive shell,

docker run -it d9798e275eb3 bash

To list out the docker process type – ***docker ps -a***



```
netlab@netlab:~/code/docker$ sudo docker run d9798e275eb3
netlab@netlab:~/code/docker$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND   CREATED         STATUS                     PORTS     NAMES
f777e04f9688   d9798e275eb3   "bash"    5 seconds ago   Exited (0) 4 seconds ago             blissful_franklin
netlab@netlab:~/code/docker$ sudo docker run -it d9798e275eb3 /bin/sh
# pwd
/
# ls
bin  boot  capirca  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
# cd home
# ls
# cd ..
# cd
# ls
# git clone https://github.com/1982League/FYP-2022.git
Cloning into 'FYP-2022'...
```

Figure 6 Linux - Docker Run and process

Make sure after cloning the tool, change the permission of all the files to executable, ***chmod +x *.py***

ACL Tool Setup on Docker - MAC OS

From the directory,

sudo docker build .

docker run -it (Docker Image ID) /bin/bash

To list out the docker process type – ***docker ps -a***

Unfortunately, we don't have MAC OS, so we could not show you the screenshot.