Name: Mosab Fathy Ramadan Mohamed
Group: B20-SD-01
Lab 12: Docker

1. Compare and contrast `ENTRYPOINT` and `CMD` in Dockerfile. In what situation would you use each of them?

Answer:
CMD commands are ignored by Daemon when there are parameters stated within the docker run command while ENTRYPOINT instructions are not ignored but instead are appended as command line parameters by treating those as arguments of the command.
CMD: Sets default parameters than can be overridden from Docker Command Line Interface when a container is running
ENTRYPOINT: Sets default parameters that can't be overridden from Docker Command Line Interface when a container is running.

2. List five security precautions you will take when building or deploying a Docker resource (image or container).

Answer:
1- Keep Host and Docker Up to Date:

It is essential to patch both Docker Engine and the underlying host operating system running Docker, to prevent a range of known vulnerabilities, many of which can result in container espaces.
Since the kernel is shared by the container and the host, kernel exploits when an attacker manages to run on a container can directly affect the host. For example, a successful kernel exploit can enable attackers to break out of a non-privileged container and gain root access to the host.

2- Do Not Expose the Docker Daemon Socket:

The Docker daemon socket is a Unix network socket that facilitates communication with the Docker API. By default, this socket is owned by the root user. If anyone else obtains access to the socket, they will have permissions equivalent to root access to the host.

3- Run Docker in Rootless Mode

Docker provides "rootless mode", which lets you run Docker daemons and containers as non-root users. This is extremely important to mitigate vulnerabilities in daemons and container runtimes, which can grant root access of entire nodes and clusters to an attacker.

4- Avoid Privileged Containers

Docker provides a privileged mode, which lets a container run as root on the local machine. Running a container in privileged mode provides the capabilities of that host—including:

- Root access to all devices
- Ability to tamper with Linux security modules like AppArmor and SELinux
- Ability to install a new instance of the Docker platform, using the host's kernel capabilities, and run Docker within Docker.

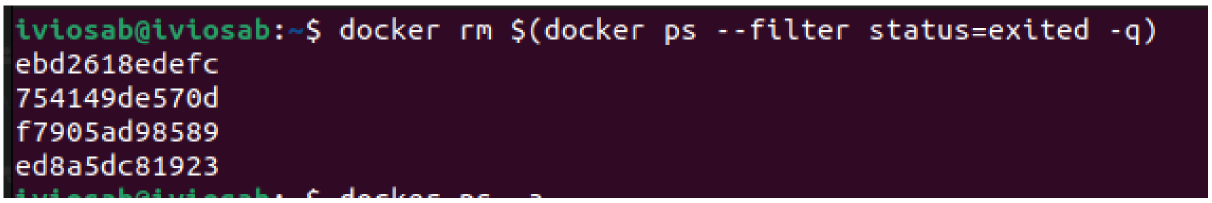
Privileged containers create a major security risk—enabling attackers to easily escalate privileges if the container is compromised. Therefore, it is not recommended to use privileged containers in a production environment. Best of all, never use them in any environment.

5- Limit Container Resources

When a container is compromised, attackers may try to make use of the underlying host resources to perform malicious activity. Set Docker memory and CPU usage limits to minimize the impact of breaches for resource-intensive containers.

3. Show a single line command that will remove all exited Docker containers. Do not use any text filtering editor. Show test results.

Answer:

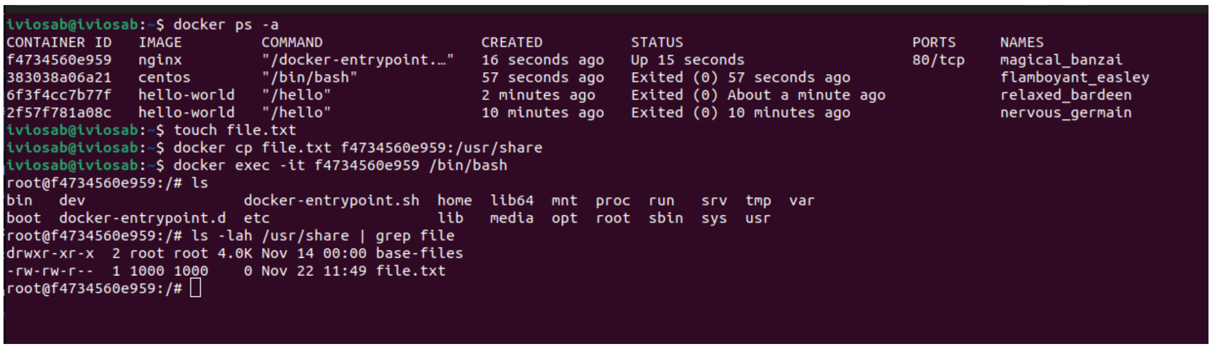


```
iviosab@iviosab:~$ docker rm $(docker ps --filter status=exited -q)
ebd2618edefc
754149de570d
f7905ad98589
ed8a5dc81923
iviosab@iviosab:~$ docker ps -a
```

4. Show how you can copy files to a running container without entering the container's interactive shell.

Answer:

We can create a txt file then run the docker image then get the container id and run “docker cp something.txt containerId:PATH”

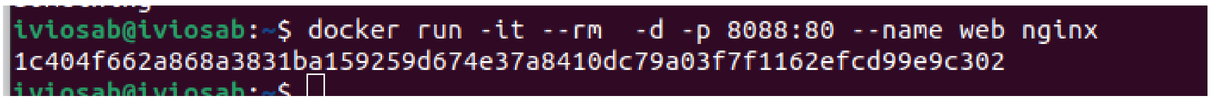


```
iviosab@iviosab:~$ docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS                        PORTS     NAMES
f4734560e959   nginx         "/docker-entrypoint.…"   16 seconds ago   Up 15 seconds                 80/tcp    magical_banzai
383038a06a21   centos        "/bin/bash"              57 seconds ago   Exited (0) 57 seconds ago               flamboyant_easley
6f3f4cc7b77f   hello-world   "/hello"                 2 minutes ago    Exited (0) About a minute ago           relaxed_bardeen
2f57f781a08c   hello-world   "/hello"                 10 minutes ago   Exited (0) 10 minutes ago               nervous_germain
iviosab@iviosab:~$ touch file.txt
iviosab@iviosab:~$ docker cp file.txt f4734560e959:/usr/share
iviosab@iviosab:~$ docker exec -it f4734560e959 /bin/bash
root@f4734560e959:/# ls
bin   dev                  docker-entrypoint.sh  home  lib64  mnt  proc  run   srv  tmp  var
boot  docker-entrypoint.d  etc                   lib   media  opt  root  sbin  sys  usr
root@f4734560e959:/# ls -lah /usr/share | grep file
drwxr-xr-x  2 root root 4.0K Nov 14 00:00 base-files
-rw-rw-r--  1 1000 1000    0 Nov 22 11:49 file.txt
root@f4734560e959:/# 
```

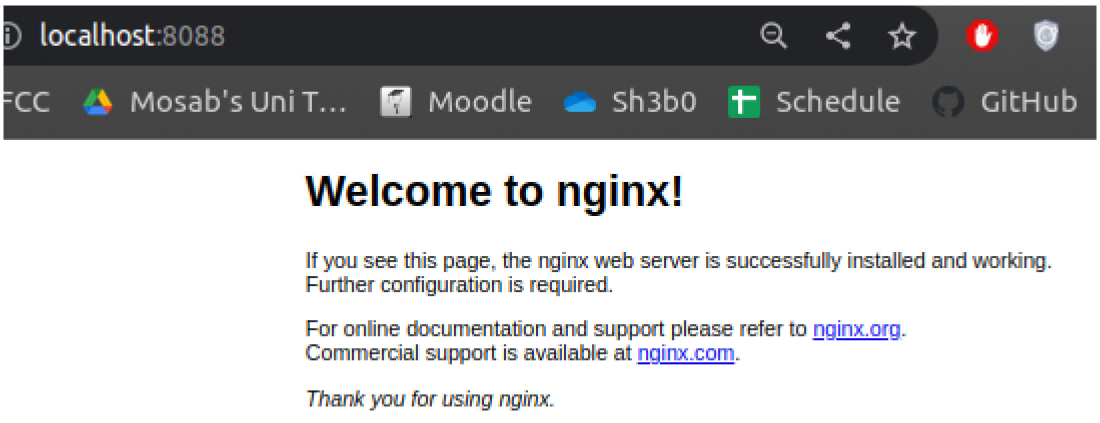
5. Create a dockerized web application running on nginx. The web index page `index.html` should be located on your host machine. The directory containing the index page should be mounted to the container and served from there.

> This means that you should be able to modify the web index page on your host machine without interacting with the container.
> Show all steps taken for the configuration including the test results.

Answer:



```
iviosab@iviosab:~$ docker run -it --rm  -d -p 8088:80 --name web nginx
1c404f662a868a3831ba159259d674e37a8410dc79a03f7f1162efcd99e9c302
iviosab@iviosab:~$ 
```

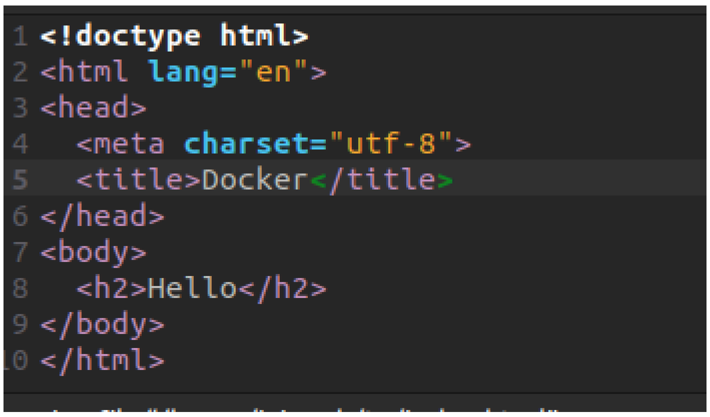


Welcome to nginx!

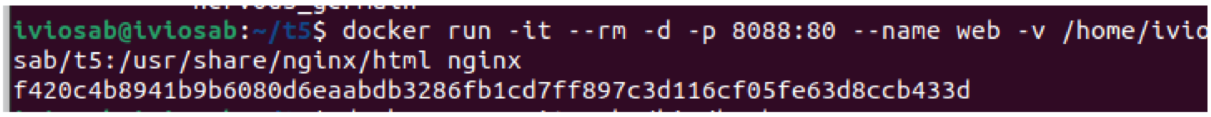
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

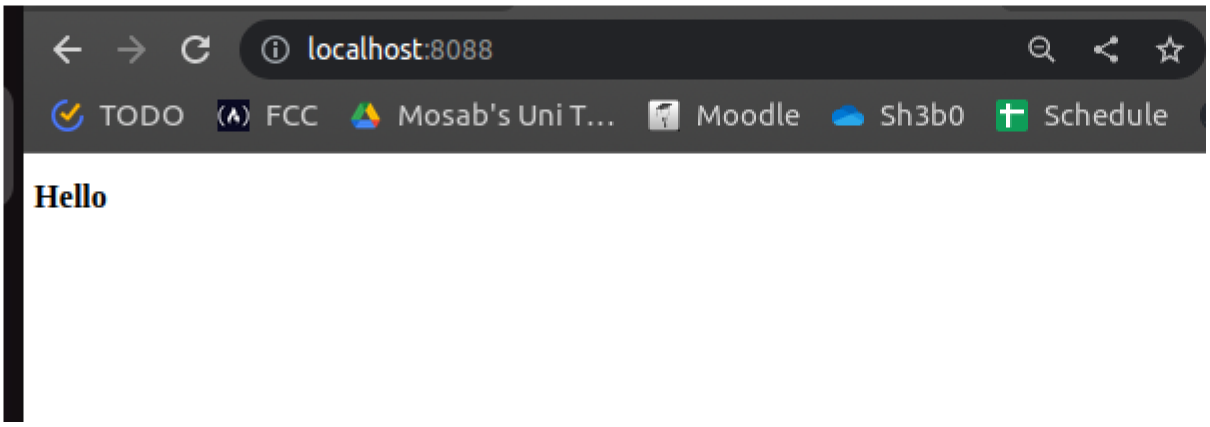
Thank you for using nginx.



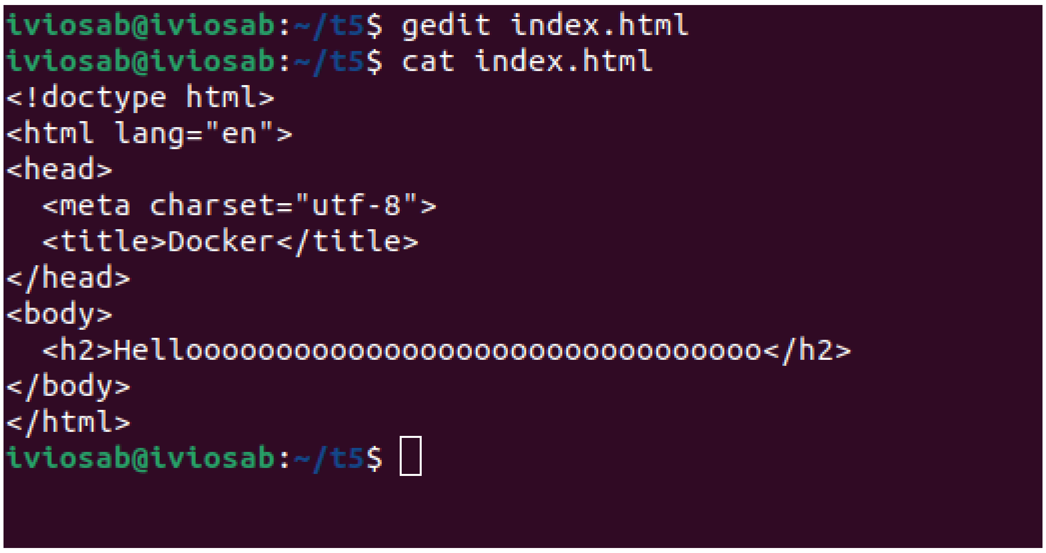
```html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Docker</title>
</head>
<body>
  <h2>Hello</h2>
</body>
</html>
```



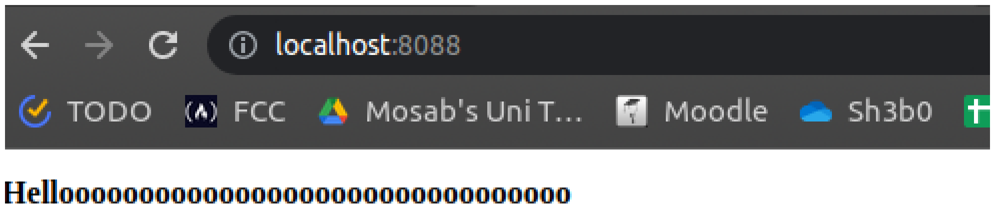
```
iviosab@iviosab:~/t5$ docker run -it --rm -d -p 8088:80 --name web -v /home/ivio
sab/t5:/usr/share/nginx/html nginx
f420c4b8941b9b6080d6eaabdb3286fb1cd7ff897c3d116cf05fe63d8ccb433d
```



Hello



```
iviosab@iviosab:~/t5$ gedit index.html
iviosab@iviosab:~/t5$ cat index.html
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Docker</title>
</head>
<body>
  <h2>Helloooooooooooooooooooooooooooooooooo</h2>
</body>
</html>
iviosab@iviosab:~/t5$ 
```

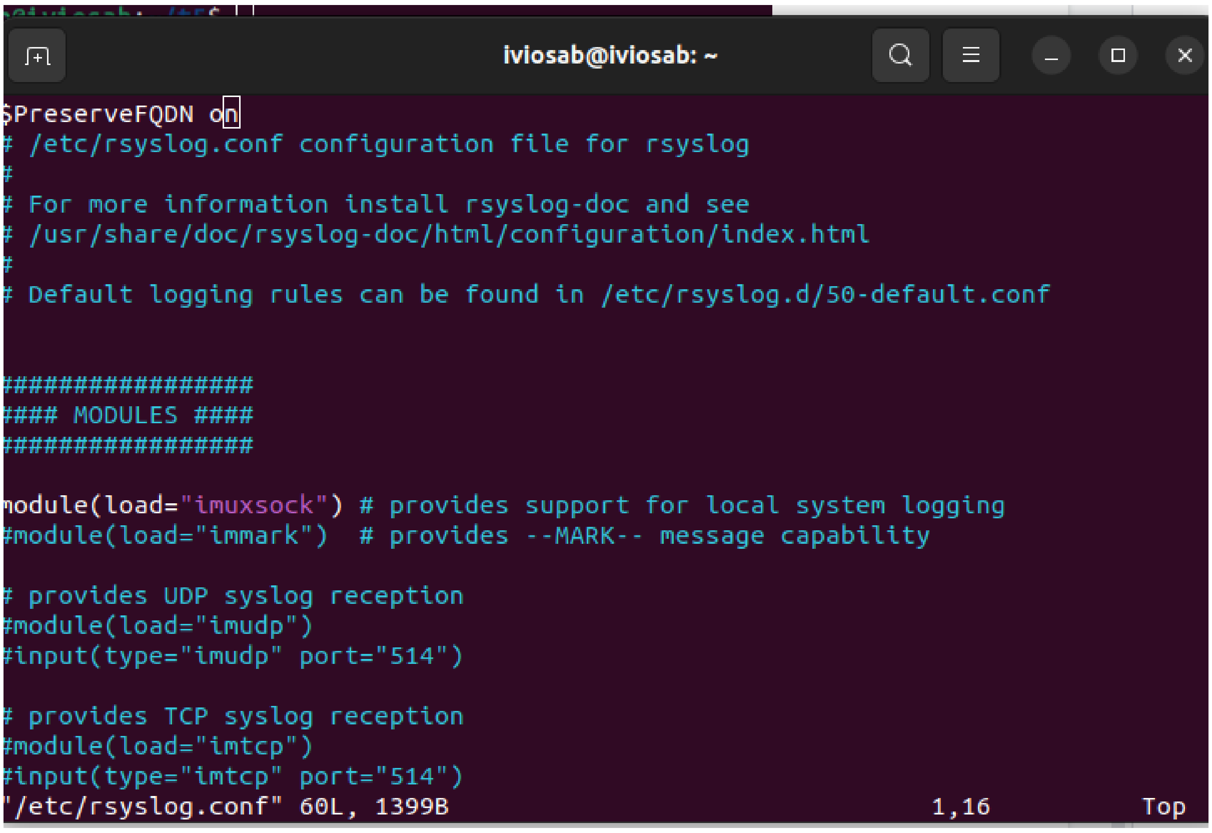


Helloooooooooooooooooooooooooooooooooo

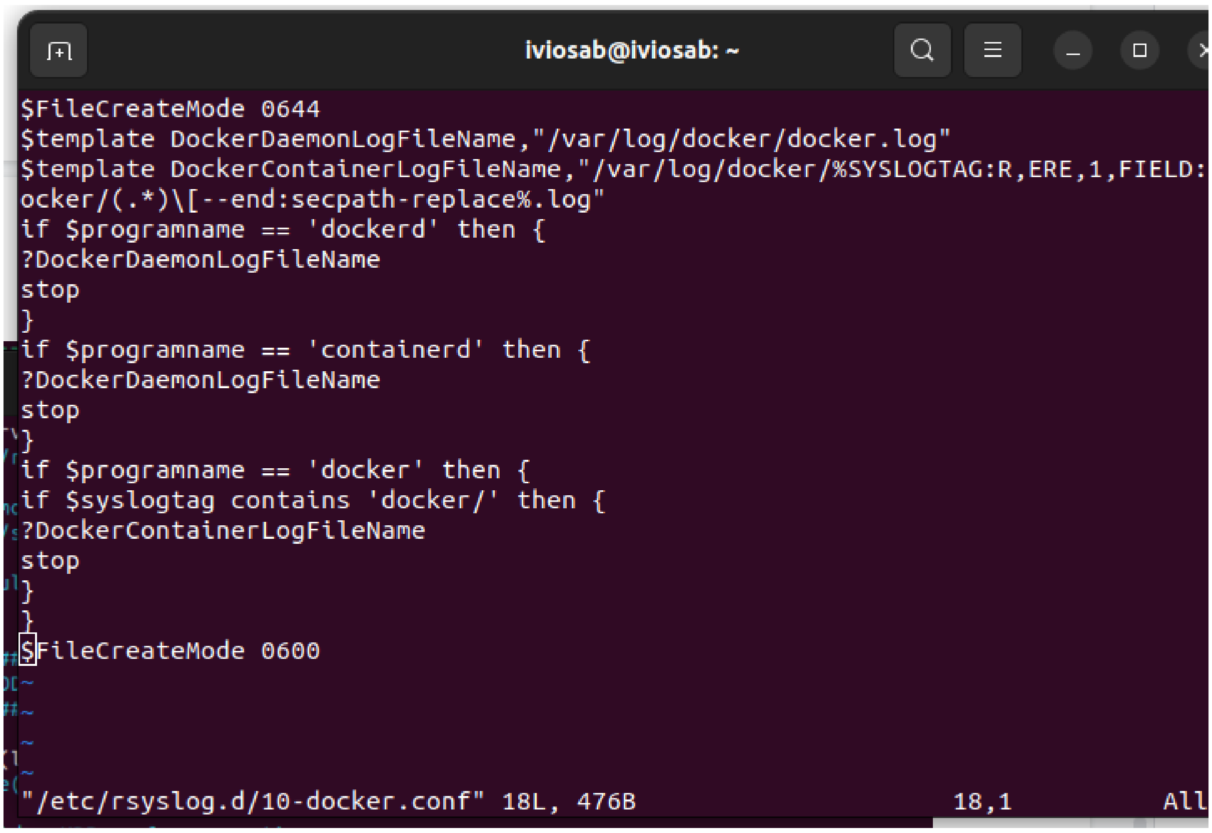
6. Setup rsyslog on your host machine as a central logging server. Create a Docker container and configure it to forward its log to your central logging server.

   > Show steps and test results.

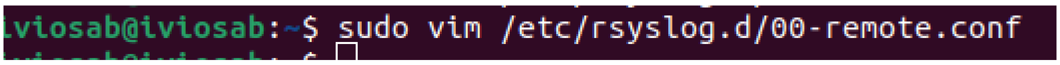
Answer:



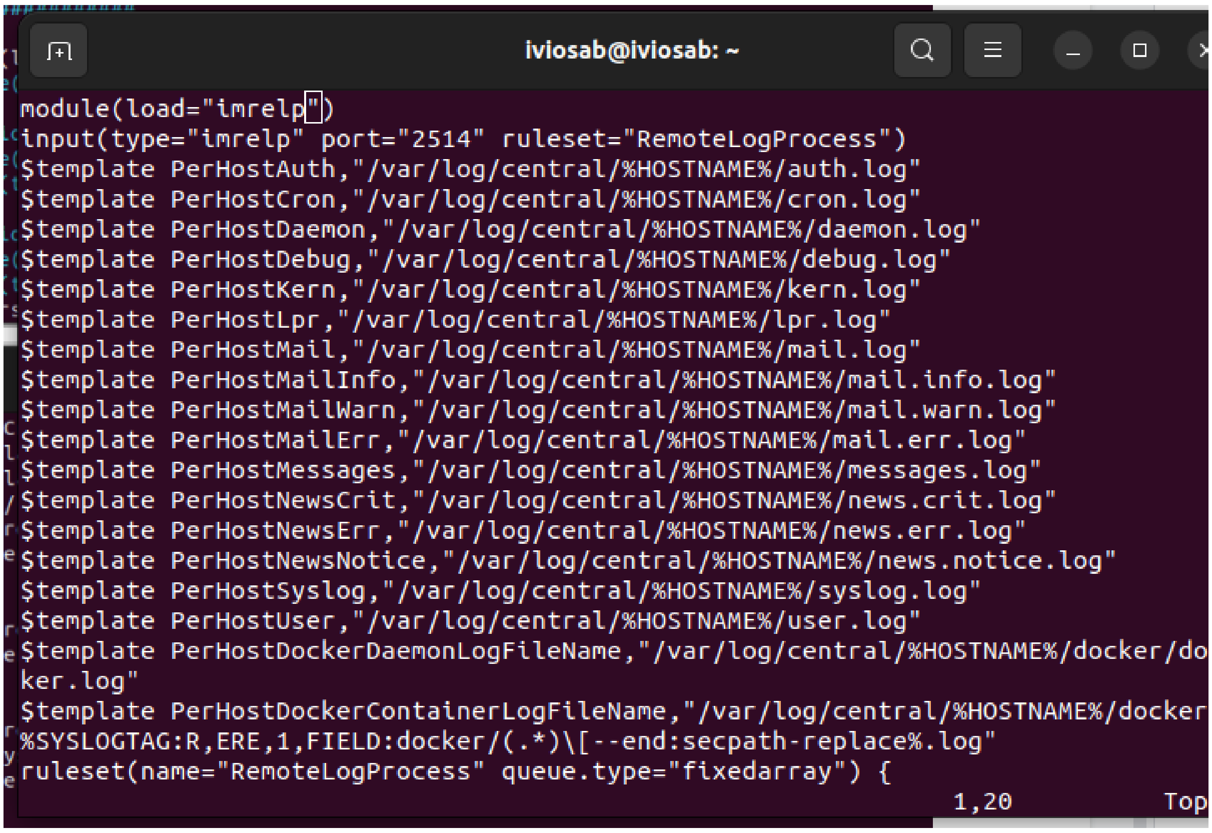
```
$PreserveFQDN on
# /etc/rsyslog.conf configuration file for rsyslog
#
# For more information install rsyslog-doc and see
# /usr/share/doc/rsyslog-doc/html/configuration/index.html
#
# Default logging rules can be found in /etc/rsyslog.d/50-default.conf


#################
#### MODULES ####
#################

module(load="imuxsock") # provides support for local system logging
#module(load="immark")  # provides --MARK-- message capability

# provides UDP syslog reception
#module(load="imudp")
#input(type="imudp" port="514")

# provides TCP syslog reception
#module(load="imtcp")
#input(type="imtcp" port="514")
"/etc/rsyslog.conf" 60L, 1399B                                1,16          Top
```



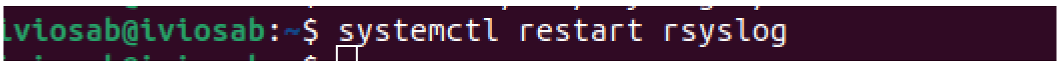
```
$FileCreateMode 0644
$template DockerDaemonLogFileName,"/var/log/docker/docker.log"
$template DockerContainerLogFileName,"/var/log/docker/%SYSLOGTAG:R,ERE,1,FIELD:
ocker/(.*)\[--end:secpath-replace%.log"
if $programname == 'dockerd' then {
?DockerDaemonLogFileName
stop
}
if $programname == 'containerd' then {
?DockerDaemonLogFileName
stop
}
if $programname == 'docker' then {
if $syslogtag contains 'docker/' then {
?DockerContainerLogFileName
stop
}
}
$FileCreateMode 0600
~
~
~
~
"/etc/rsyslog.d/10-docker.conf" 18L, 476B                     18,1          All
```



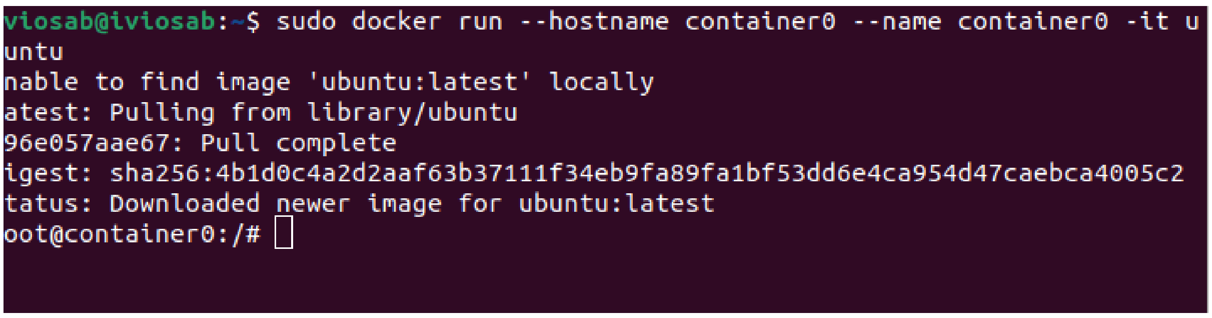
```
iviosab@iviosab:~$ sudo vim /etc/rsyslog.d/00-remote.conf
```



```
module(load="imrelp")
input(type="imrelp" port="2514" ruleset="RemoteLogProcess")
$template PerHostAuth,"/var/log/central/%HOSTNAME%/auth.log"
$template PerHostCron,"/var/log/central/%HOSTNAME%/cron.log"
$template PerHostDaemon,"/var/log/central/%HOSTNAME%/daemon.log"
$template PerHostDebug,"/var/log/central/%HOSTNAME%/debug.log"
$template PerHostKern,"/var/log/central/%HOSTNAME%/kern.log"
$template PerHostLpr,"/var/log/central/%HOSTNAME%/lpr.log"
$template PerHostMail,"/var/log/central/%HOSTNAME%/mail.log"
$template PerHostMailInfo,"/var/log/central/%HOSTNAME%/mail.info.log"
$template PerHostMailWarn,"/var/log/central/%HOSTNAME%/mail.warn.log"
$template PerHostMailErr,"/var/log/central/%HOSTNAME%/mail.err.log"
$template PerHostMessages,"/var/log/central/%HOSTNAME%/messages.log"
$template PerHostNewsCrit,"/var/log/central/%HOSTNAME%/news.crit.log"
$template PerHostNewsErr,"/var/log/central/%HOSTNAME%/news.err.log"
$template PerHostNewsNotice,"/var/log/central/%HOSTNAME%/news.notice.log"
$template PerHostSyslog,"/var/log/central/%HOSTNAME%/syslog.log"
$template PerHostUser,"/var/log/central/%HOSTNAME%/user.log"
$template PerHostDockerDaemonLogFileName,"/var/log/central/%HOSTNAME%/docker/do
ker.log"
$template PerHostDockerContainerLogFileName,"/var/log/central/%HOSTNAME%/docker
%SYSLOGTAG:R,ERE,1,FIELD:docker/(.*)\[--end:secpath-replace%.log"
ruleset(name="RemoteLogProcess" queue.type="fixedarray") {
                                                          1,20          Top
```



```
iviosab@iviosab:~$ systemctl restart rsyslog
```



```
viosab@iviosab:~$ sudo docker run --hostname container0 --name container0 -it u
untu
nable to find image 'ubuntu:latest' locally
atest: Pulling from library/ubuntu
96e057aae67: Pull complete
igest: sha256:4b1d0c4a2d2aaf63b37111f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2
tatus: Downloaded newer image for ubuntu:latest
oot@container0:/# 
```

7. Dockerize any open source application of your choice, and host it on Docker hub. Share link to the repository.

Answer:

8. Find and fix the problems in the following Dockerfile. There are some issues building the image and also running the container:

   ```
   FROM alpine
   RUN apt-get update && apt-get install -y python3 --no-install-recommends
   RUN touch index.html
   RUN echo "<html><h1>Testing web</h1></html>" >> index.html
   CMD ["python", "-m", "http.server"]
   ```

   > Show all steps taken to fix it, and a working solution.

Answer:

“

```
FROM alpine:3.17

WORKDIR /app

RUN apk update && apk add --no-cache python3=~3.10 \
    && touch index.html \
    && echo "<html><h1>Testing web</h1></html>" >> index.html \
    && addgroup -S app && adduser -S app -G app \
    && chown -R app:app .

EXPOSE 8000

USER app

CMD ["python3", "-m", "http.server"]
```

“