

[illegible]

| | |
|-------------|---|
| Name | Dockerlint |
| URL | https://attackdefense.com/challengedetails?cid=2163 |
| Type | Docker Security: Dockerfile Linting |

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Dockerfile Linting is a process to check and modify the Dockerfile as per the industry's best practices.

[Dockerlint](#) is a linting tool by RedCoolBeans based on recommendations from [Dockerfile Reference](#) and [Best practices for writing Dockerfiles](#).

A Dockerfile is provided in the home directory of the root user (i.e. /root). Dockerlint is installed on the machine.

Objective: Analyze the Dockerfile with Dockerlint. Edit Dockerfile to remove all issues detected by Dockerlint!

Solution:

Step 1: List files present in current directory.

Command: ls

```
root@attackdefense:~#  
root@attackdefense:~# ls  
Dockerfile  
root@attackdefense:~#
```

There is a Dockerfile present in the current directory.

Step 2: Check help menu for dockerlint.

Command: dockerlint -h

```
root@attackdefense:~# dockerlint -h
Dockerlint 0.3.11

      usage: dockerlint [-h] [-d] [-p] [-f Dockerfile]
root@attackdefense:~#
```

Step 3: Read the contents of Dockerfile.

Command: cat -n Dockerfile

```
root@attackdefense:~# cat -n Dockerfile
 1 FROM debian:
 2 MAINTAINER maintainer@redcoolbeans.com
 3
 4 RUN apt-get update && apt-get -y install nodejs
 5
 6 COPY package.json usr/src/app
 7
 8 RUN cd /usr/src/app \
 9 && npm install node-static
10
11 RUN sudo ls /root
12
13 ADDD http://examplesite.org/example.tar .
14
15 EXPOSE 8000
16 CMD npm start
```

Step 4: Run dockerlint on Dockerfile.

Command: dockerlint -f Dockerfile

```
root@attackdefense:~# dockerlint -f Dockerfile
ERROR: Tag must not be empty for "debian" on line 1
WARN: Recommended exec/array form not used on line 16
WARN: sudo(8) usage found on line 11 which is discouraged
ERROR: Lines cannot have trailing spaces
ERROR: ADDD is invalid on line 13
WARN: MAINTAINER instruction is deprecated on line 2

ERROR: Dockerfile failed.
```

Explanation

Tag must not be empty: While defining the base image, a particular tag should be used to avoid problems faced in future build. 'latest' tag can be used but is not advised as it changes as the developer pushes updates to the base image.

Recommended exec/array form not used: ENTRYPOINT and CMD instructions should be used with array form of commands. When using the plain text version of passing arguments, signals from the OS are not correctly passed to the executables.

sudo(8) usage found on line 10 which is discouraged: You should avoid installing or using 'sudo' since it has unpredictable TTY and signal-forwarding behavior that can cause more problems than it solves.

Lines cannot have trailing spaces: Trailing space in lines is considered as bad practice and should be avoided while writing Dockerfiles.

ADDD is invalid: Only valid commands are allowed in a Dockerfile. This seems to be a typo of ADD command.

MAINTAINER instruction is deprecated: This INSTRUCTION is deprecated as of Docker 1.13 and can be replaced by using LABEL maintainer="value".

Modify the Dockerfile to address these issues.

Please note that line numbers below are respective to unmodified Dockerfile.

Modifications:

Line 1: Specify tag for the base image used.

Before Modification: FROM debian:

After Modification: FROM debian:9

Line 2: Use LABEL for specifying maintainer.

Before Modification: MAINTAINER maintainer@redcoolbeans.com

After Modification: LABEL maintainer="maintainer@redcoolbeans.com"

Line 6: Remove trailing space from the COPY Command.

Before Modification: COPY package.json usr/src/app

After Modification: COPY package.json usr/src/app

Line 11: Remove sudo from command.

Before Modification: RUN sudo ls /root

After Modification: RUN ls /root

Line 13: Correct the spelling of ADD statement.

Before Modification: ADDD http://examplesite.org/example.tar .

After Modification: ADD http://examplesite.org/example.tar .

Line 16: Use recommended exec/array form.

Before Modification: CMD npm start

After Modification: CMD ["npm", "start"]

Step 5: Check the file in nano after applying the above mentioned modifications.

Command: nano -l Dockerfile

```
GNU nano 2.9.3 Dockerfile
1 FROM debian:9
2 LABEL maintainer="maintainer@redcoolbeans.com"
3
4 RUN apt-get update && apt-get -y install nodejs
5
6 COPY package.json usr/src/app
7
8 RUN cd /usr/src/app \
9 && npm install node-static
10
11 RUN ls /root
12
13 ADD http://examplesite.org/example.tar .
14
15 EXPOSE 8000
16 CMD ["npm", "start"]
17
```

Save the file and exit nano. Press 'Ctrl + X' followed by 'Y' and Enter to exit and save changes.

Step 6: Run dockerlint again on the modified Dockerfile.

Command: dockerlint -f Dockerfile

```
root@attackdefense:~# dockerlint -f Dockerfile
INFO: Dockerfile is OK.
root@attackdefense:~#
```

Dockerfile is now OK and free from errors/issues.

References:

- dockerlint (<https://github.com/RedCoolBeans/dockerlint>)

