



Lab Experiment – 8

NAME -Divyaansh Jain

ROLL NO – R171218040

SAP ID – 500067134

COURSE – B. Tech CSE- DevOps Batch 1

SUBJECT – Application Containerization

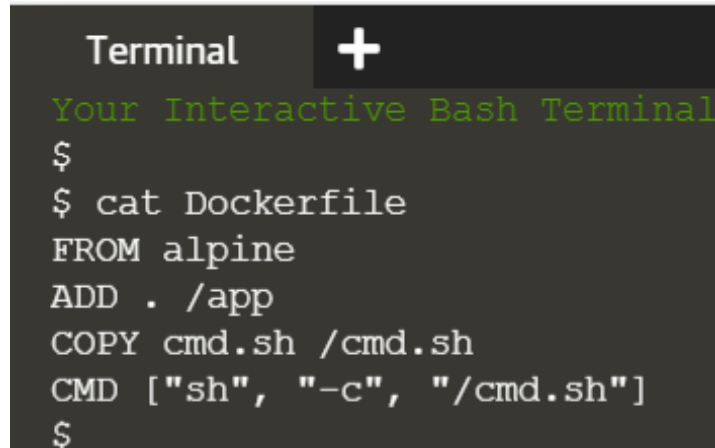
SEMESTER – 6th semester

Submitted To:

Dr. Hitesh Kumar Sir

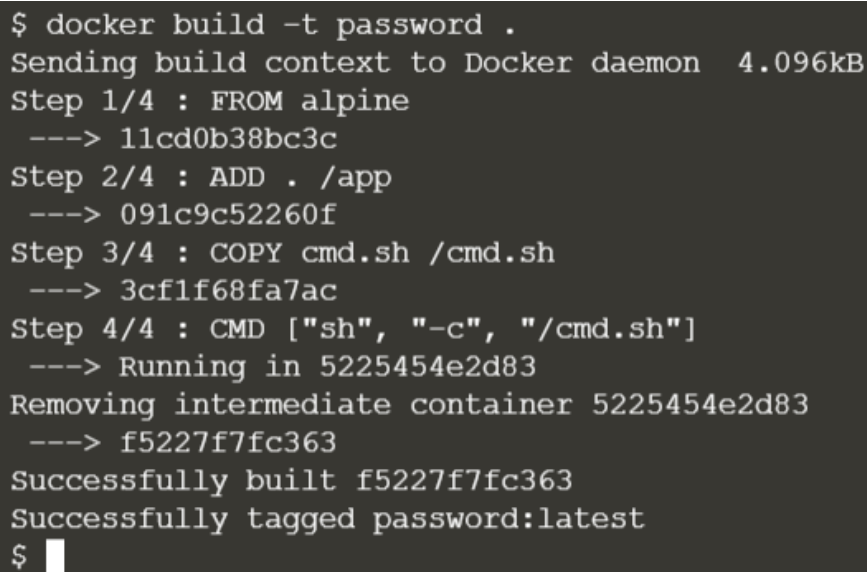
Docker Ignore

- Use the cat command on the Dockerfile to see inside the steps. Currently our app folder contains password.txt file which contains sensitive information.

A terminal window titled "Terminal" with a "+" icon. The text "Your Interactive Bash Terminal" is displayed in green. The terminal shows the command "\$ cat Dockerfile" and its output: "FROM alpine", "ADD . /app", "COPY cmd.sh /cmd.sh", "CMD [\"sh\", \"-c\", \"/cmd.sh\"]", and a prompt "\$".

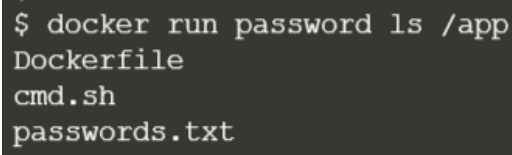
```
Terminal +
Your Interactive Bash Terminal
$
$ cat Dockerfile
FROM alpine
ADD . /app
COPY cmd.sh /cmd.sh
CMD ["sh", "-c", "/cmd.sh"]
$
```

- Using the docker build command to build this Dockerfile.

A terminal window showing the output of the "docker build -t password ." command. It details the build steps: FROM alpine, ADD . /app, COPY cmd.sh /cmd.sh, and CMD ["sh", "-c", "/cmd.sh"], along with intermediate container IDs and the final image ID f5227f7fc363.

```
$ docker build -t password .
Sending build context to Docker daemon 4.096kB
Step 1/4 : FROM alpine
----> 11cd0b38bc3c
Step 2/4 : ADD . /app
----> 091c9c52260f
Step 3/4 : COPY cmd.sh /cmd.sh
----> 3cf1f68fa7ac
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
----> Running in 5225454e2d83
Removing intermediate container 5225454e2d83
----> f5227f7fc363
Successfully built f5227f7fc363
Successfully tagged password:latest
$
```

- If we create a container from the image created then we will be able to see the password.txt.

A terminal window showing the output of the "docker run password ls /app" command. The output lists the files in the /app directory: "Dockerfile", "cmd.sh", and "passwords.txt".

```
$ docker run password ls /app
Dockerfile
cmd.sh
passwords.txt
$
```

- To ignore this file, we can use `.dockerignore`. Create a `.dockerignore` file and inside this file write those files name which you want to ignore when building the docker image. Now when we build the image, we will see the container will not contain `password.txt` file.

```
$ echo passwords.txt >> .dockerignore
$ docker build -t nopassword .
Sending build context to Docker daemon 104.9MB
Step 1/4 : FROM alpine
---> 11cd0b38bc3c
Step 2/4 : ADD . /app
---> f81e2760b406
Step 3/4 : COPY cmd.sh /cmd.sh
docker run nopassword ls /app
---> 19c6392ef945
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
---> Running in 479982d2c2c3
Removing intermediate container 479982d2c2c3
---> d687647ffb61
Successfully built d687647ffb61
Successfully tagged nopassword:latest
$ docker run nopassword ls /app
Dockerfile
big-temp-file.img
cmd.sh
$
```

- Using the `.dockerignore` we can also be used to improve the build time of images. For example, suppose an app folder contains a temporary file. Since these files are not required further, these will simply increase build time.

```
$ docker build -t large-file-context .
Sending build context to Docker daemon 104.9MB
Step 1/4 : FROM alpine
---> 11cd0b38bc3c
Step 2/4 : ADD . /app
---> Using cache
---> cd1a5d44625b
Step 3/4 : COPY cmd.sh /cmd.sh
---> Using cache
---> a38388cad43a
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
---> Using cache
---> a7e7ae6dfc20
Successfully built a7e7ae6dfc20
```

- We can use `.dockerignore` file to exclude files which we don't want to be sent to the Docker Build Context during the build. To speed up our build, simply include the filename of the large file in the ignore file. When we rebuild the image, it will be much faster as it doesn't have to copy the 100M file.

```
$ echo big-temp-file.img >> .dockerignore
$ docker build -t no-large-file-context .
Sending build context to Docker daemon 4.096kB
Step 1/4 : FROM alpine
---> 11cd0b38bc3c
Step 2/4 : ADD . /app
---> 36a9c93f1ec8
Step 3/4 : COPY cmd.sh /cmd.sh
---> f57660b31195
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
---> Running in 8077be293812
Removing intermediate container 8077be293812
---> 00325c7a4454
Successfully built 00325c7a4454
Successfully tagged no-large-file-context:latest
$
```