Anurag Pachauri
500069505
R171218026

Implementing Docker ignore.

Procedure:

1) Use command **cat Dockerfile** to see inside the steps of Dockerfile. Our app folder contains password.txt file which is sensitive.

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

2) Use command **docker build** to build this Dockerfile.

```
$ docker build -t password .
Sending build context to Docker daemon  104.9MB
Step 1/4 : FROM alpine
 ---> 11cd0b38bc3c
Step 2/4 : ADD . /app
 ---> f0a2408880c4
Step 3/4 : COPY cmd.sh /cmd.sh
 ---> 6d77467c0177
Step 4/4 : CMD ["sh", "-c", "/cmd.sh"]
 ---> Running in 652da9d335b3
Removing intermediate container 652da9d335b3
 ---> 11bead8c855c
Successfully built 11bead8c855c
Successfully tagged password:latest
```

3) Now if you see content by starting container from this image, you will see this also contains password.txt file.

```
$ docker run password ls /app
Dockerfile
big-temp-file.img
cmd.sh
passwords.txt
```

4) To ignore these sensitive files, we can use .dockerignore. Create a .dockerignore file and inside this file write those files name which you want to ignore when building docker image. Now if you build image you 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
$
```

5) .dockerignore can also be used to improve the build time of images. For example, suppose an app folder contains 100 mb of 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
```

6) We can use .dockerignore 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
$
```