Damir Nabiullin - Lab 12

1. Docker has a default entrypoint which is `/bin/sh -c` but does not have a default command.
   When you run docker like this: `docker run -i -t ubuntu bash` the entrypoint is the default `/bin/sh -c`, the image is `ubuntu` and the command is `bash`.
   The command is run via the entrypoint. i.e., the actual thing that gets executed is `/bin/sh -c bash`. This allowed Docker to implement `RUN` quickly by relying on the shell's parser.
   Later on, people asked to be able to customize this, so `ENTRYPOINT` and `--entrypoint` were introduced.
   Everything after the image name, `ubuntu` in the example above, is the command and is passed to the entrypoint. When using the `CMD` instruction, it is exactly as if you were executing `docker run -i -t ubuntu <cmd>`. The parameter of the entrypoint is `<cmd>`.
   You will also get the same result if you instead type this command `docker run -i -t ubuntu`: a bash shell will start in the container because in the ubuntu Dockerfile a default `CMD` is specified: `CMD ["bash"]`

2. Precautions:
   a) Keep Host and Docker Up to Date
   b) Do Not Expose the Docker Daemon Socket
   c) Run Docker in Rootless Mode
   d) Avoid Privileged Containers
   e) Limit Container Resources

3. Command: `docker rm -f $(docker ps -a -q)`

```
dale@dale-nitro:~$ docker run -d nginx
06c66bd1f153baab65ea55d12e13ad95e22d67a3a9d4323d50c96e5e5093fded
dale@dale-nitro:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS     NAMES
06c66bd1f153   nginx     "/docker-entrypoint.…"   1 second ago    Up 1 second    80/tcp    affectionate_sanderson
dale@dale-nitro:~$ docker rm -f $(docker ps -a -q)
06c66bd1f153
8a3c2cb2244b
dale@dale-nitro:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
dale@dale-nitro:~$
```

4. Command: `docker cp task4.txt intelligent_black:/usr/share/nginx/html/`

```
dale@dale-nitro:~$ docker run -d nginx
6678ec4c19049632db033903ea5dfe5d02badb5c9a7439a69fe9851aa7ee839d
^[[Adale@dale-nitro^C
dale@dale-nitro:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                      PORTS     NAMES
6678ec4c1904   nginx     "/docker-entrypoint.…"   19 seconds ago       Up 18 seconds               80/tcp    intelligent_black
a69369f8e636   nginx     "/docker-entrypoint.…"   About a minute ago   Exited (0) 23 seconds ago             intelligent_vaughan
dale@dale-nitro:~$ docker exec -it intelligent_black bash
root@6678ec4c1904:/# 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@6678ec4c1904:/# ls /usr/share/nginx/html/
50x.html  index.html
root@6678ec4c1904:/# exit
exit
dale@dale-nitro:~$ echo "Hello" > task4.txt
```

```
dale@dale-nitro:~$ docker cp task4.txt intelligent_black:/usr/share/nginx/html/
dale@dale-nitro:~$ docker exec -it intelligent_black bash
root@6678ec4c1904:/# ls /usr/share/nginx/html/
50x.html  index.html  task4.txt
root@6678ec4c1904:/# exit
exit
```

5. I used such command: `docker run -v "$(pwd):/usr/share/nginx/html" -d -p 8080:80 nginx`

```
dale@dale-nitro:~/Documents/SNA/Week 12$ docker run -v "$(pwd):/usr/share/nginx/html" -d -p 8080:80 nginx
5f1e00bf40f7f319568b494f3647b1b06130f2c83e70488994a025c0a8c26d23
dale@dale-nitro:~/Documents/SNA/Week 12$ cat index.html
<!doctype html>
<html>
  <head>
    <title>Damir Nabiullin</title>
  </head>
  <body>
    <p>Damir Nabiullin</p>
  </body>
</html>dale@dale-nitro:~/Documents/SNA/Week 12$ curl http://localhost:8080/
<!doctype html>
<html>
  <head>
    <title>Damir Nabiullin</title>
  </head>
  <body>
    <p>Damir Nabiullin</p>
  </body>
```

6. I tried different ways but I got connection refused :(

```
docker run -p 8080:80 -d --name task6-container task6
80f1cb21395eaa31d435e9b91374e31116b1d8cc9b135ffc4261665c6ac70889

docker run -it task6 bash
```

```
dale@dale-nitro:~/Documents/SNA/Week 12$ sudo nano /etc/rsyslog.conf
[sudo] password for dale:
dale@dale-nitro:~/Documents/SNA/Week 12$ nano /etc/rsyslog.d/remote.conf
dale@dale-nitro:~/Documents/SNA/Week 12$ sudo nano /etc/rsyslog.d/remote.conf
dale@dale-nitro:~/Documents/SNA/Week 12$ cat /etc/rsyslog.d/remote.conf
$template RemoteLogs,"/var/log/%HOSTNAME%/%PROGRAMNAME%.log"
*.* ?RemoteLogs
& ~
dale@dale-nitro:~/Documents/SNA/Week 12$ cat server.conf
*.* @@0.0.0.0:514dale@dale-nitro:~/Documents/SNA/Week 12$ ^C
dale@dale-nitro:~/Documents/SNA/Week 12$ cat Dockerfile
FROM ubuntu:jammy
RUN apt-get update
RUN apt-get install -y rsyslog
ADD ./server.conf /etc/rsyslog.d/server.confdale@dale-nitro:~/Documents/SNA/Week 12$ ^C
dale@dale-nitro:~/Documents/SNA/Week 12$ docker build -t task6:latest .
Sending build context to Docker daemon   5.12kB
Step 1/4 : FROM ubuntu:jammy
jammy: Pulling from library/ubuntu
e96e057aae67: Pull complete
Digest: sha256:4b1d0c4a2d2aaf63b37111f34eb9fa89fa1bf53dd6e4ca954d47caebca4005c2
Status: Downloaded newer image for ubuntu:jammy
 ---> a8780b506fa4
Step 2/4 : RUN apt-get update
 ---> Running in a519a7cab009
```

7. -

8. In this case we should replace `apt-get` with `apk`, `python` with `python3`, and `CMD` with `ENTRYPOINT`.

```
dale@dale-nitro:~/Documents/SNA/Week 12$ cat Dockerfile
FROM alpine
RUN apk update && apk add python3
RUN touch index.html
RUN echo "<html><h1>Testing web</h1></html>" >> index.html
ENTRYPOINT ["python3", "-m", "http.server"]dale@dale-nitro:~/Documents/SNA/Week 12$
dale@dale-nitro:~/Documents/SNA/Week 12$ docker build -t task8 .
Sending build context to Docker daemon   5.12kB
Step 1/5 : FROM alpine
 ---> bfe296a52501
Step 2/5 : RUN apk update && apk add python3
 ---> Using cache
```

```
dale@dale-nitro:~/Documents/SNA/Week 12$ docker run -p 8080:80 -d --name task8-container task8
739242170d612ecd07c444f21449488114e25fcddbb8e5c970338cf366ab1204
```