

CLOUD MANAGEMENT

LAB #5: CONTAINERS

JOVITA ANDREWS

Part 1:

What is Docker, and how is it related to containers?

- Docker is a platform that allows developers to create, deploy, and manage applications in containers. Containers are lightweight, standalone packages of software that include everything needed to run an application: code, runtime, system tools, and libraries. Docker provides an easy way to package and distribute applications across various environments consistently, as it abstracts away the specifics of the host operating system.

What is a container image?

- A container image is a standalone, executable package that includes all the necessary components (such as code, runtime, libraries, and environment variables) to run an application. Container images serve as templates for containers; when an image is instantiated, it becomes a running container.

What is a Dockerfile?

- A Dockerfile is a text file that contains a set of instructions for assembling a Docker image. It describes the steps needed to create an image, including specifying the base image, installing necessary packages, copying files, and defining environment variables and commands to run within the container.

What does the docker build command do?

- The docker build command reads the instructions from a Dockerfile and assembles a Docker image based on those instructions. When the command is run in the directory containing the Dockerfile, it generates an image that can then be used to create containers.

What is a web app, and why are they useful?

- A web app is an application that users interact with through a web browser. It operates on a web server and processes data over the internet. Web apps are useful because they are accessible from anywhere with an internet connection, are platform-independent, and require no installation on the user's device, making them highly accessible and easy to maintain.

Part 2:

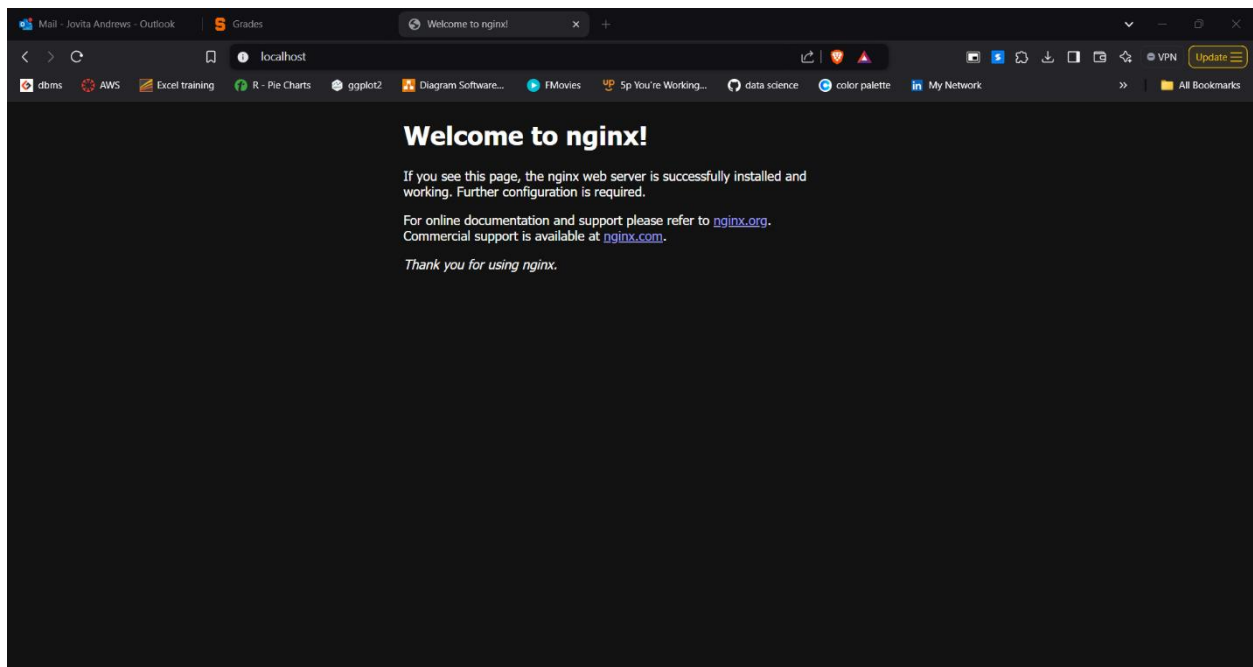
SS01: Screenshot of ls -l and uname -a commands

Description: This screenshot shows the output of the `ls -l` command, listing files and directories within the root directory of the container, and `uname -a`, which displays system information about the container, confirming it's an Ubuntu environment.

```
root@1217a2a75d5c: /  
jovita06@Jovita:~$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES  
9dcb9e8b5282   learn-databases-master-adminer:latest  "entrypoint.sh docke..." 11 minutes ago Up 55 seconds 8080/tcp     lucid_gauss  
jovita06@Jovita:~$ docker run -it ubuntu bash  
root@1217a2a75d5c:/# ls -l  
total 48  
lrwxrwxrwx   1 root root    7 Apr 22  2024 bin -> usr/bin  
drwxr-xr-x   2 root root 4096 Apr 22  2024 boot  
drwxr-xr-x   5 root root 360 Oct 29 19:15 dev  
drwxr-xr-x   1 root root 4096 Oct 29 19:15 etc  
drwxr-xr-x   3 root root 4096 Oct 11 02:09 home  
lrwxrwxrwx   1 root root    7 Apr 22  2024 lib -> usr/lib  
lrwxrwxrwx   1 root root    9 Apr 22  2024 lib64 -> usr/lib64  
drwxr-xr-x   2 root root 4096 Oct 11 02:03 media  
drwxr-xr-x   2 root root 4096 Oct 11 02:03 mnt  
drwxr-xr-x   2 root root 4096 Oct 11 02:03 opt  
dr-xr-xr-x 388 root root    0 Oct 29 19:15 proc  
drwx----- 2 root root 4096 Oct 11 02:09 root  
drwxr-xr-x   4 root root 4096 Oct 11 02:09 run  
lrwxrwxrwx   1 root root    8 Apr 22  2024 sbin -> usr/sbin  
drwxr-xr-x   2 root root 4096 Oct 11 02:03 srv  
dr-xr-xr-x  11 root root    0 Oct 29 19:15 sys  
drwxrwxrwt   2 root root 4096 Oct 11 02:09 tmp  
drwxr-xr-x  12 root root 4096 Oct 11 02:03 usr  
drwxr-xr-x  11 root root 4096 Oct 11 02:09 var  
root@1217a2a75d5c:/# uname -a  
Linux 1217a2a75d5c 5.15.153.1-microsoft-standard-WSL2 #1 SMP Fri Mar 29 23:14:13 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux  
root@1217a2a75d5c:/#
```

SS02 Screenshot of Nginx Welcome Page:

Description: After running the Nginx container, this screenshot captures the default Nginx welcome page accessed via a web browser. It confirms that the Nginx server is running on port 80 of your VM and is accessible through your host's browser.



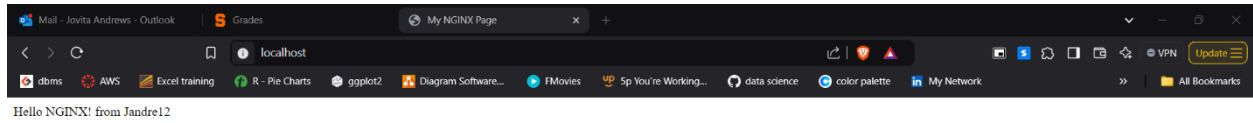
SS03: Screenshot of docker ps Showing Running Nginx Container

Description: This screenshot shows the output of `docker ps`, listing active Docker containers. It verifies that the Nginx container is running in detached mode with the specified port mapping (80:80) and assigned name (my-nginx).

```
jovita06@Jovita: ~
drwxr-xr-x  2 root root 4096 Oct 11 02:03 media
drwxr-xr-x  2 root root 4096 Oct 11 02:03 mnt
drwxr-xr-x  2 root root 4096 Oct 11 02:03 opt
dr-xr-xr-x 388 root root  0 Oct 29 19:15 proc
drwx----- 2 root root 4096 Oct 11 02:09 root
drwxr-xr-x  4 root root 4096 Oct 11 02:09 run
lrwxrwxrwx  1 root root  8 Apr 22  2024 sbin -> usr/sbin
drwxr-xr-x  2 root root 4096 Oct 11 02:03 srv
dr-xr-xr-x 11 root root  0 Oct 29 19:15 sys
drwxrwxrwt  2 root root 4096 Oct 11 02:09 tmp
drwxr-xr-x 12 root root 4096 Oct 11 02:03 usr
drwxr-xr-x 11 root root 4096 Oct 11 02:09 var
root@1217a2a75d5c:/# uname -a
Linux 1217a2a75d5c 5.15.153.1-microsoft-standard-WSL2 #1 SMP Fri Mar 29 23:14:13 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
root@1217a2a75d5c:/# exit
exit
jovita06@Jovita:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
a488a496ba95: Pull complete
f3acelb8ce45: Pull complete
11d6fdd0e8a7: Pull complete
f1091da6fd5c: Pull complete
40eea07b53d8: Pull complete
6476794e50f4: Pull complete
70850b3ec6b2: Pull complete
Digest: sha256:28402db69fec7c17e179ea87882667f1e054391138f77ffa0c3eb388efc3ffb
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
jovita06@Jovita:~$ docker run -d --name my-nginx -p 80:80 nginx
c356767a93c311547f1737d69028f00198f87ef95546cf3b9b00b721d14a871e
jovita06@Jovita:~$ docker ps
CONTAINER ID   IMAGE     STATUS   PORTS                               COMMAND
NAMES
c356767a93c3   nginx    Up       0.0.0.0:80->80/tcp                  /docker-entrypoint...
my-nginx
jovita06@Jovita:~$
```

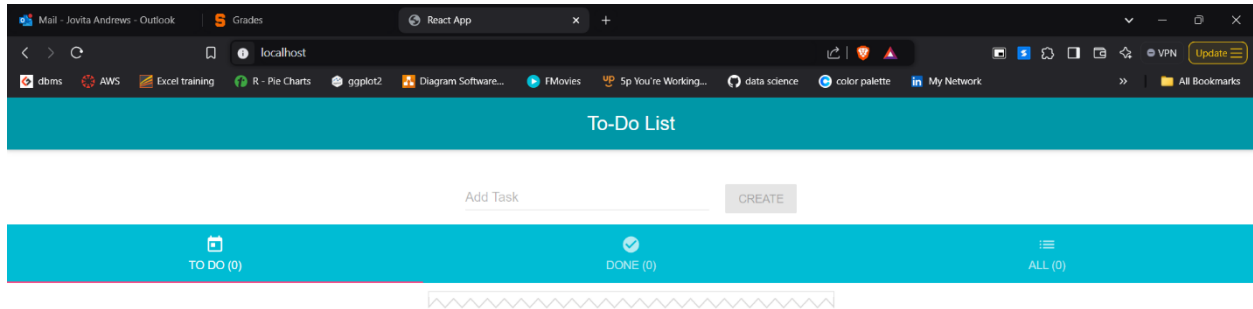
SS04: Screenshot of Customized Nginx Web Page

Description: After creating a custom Dockerfile and index.html, building the Docker image, and running the container, this screenshot captures the customized web page served by Nginx, displaying the personalized message you added in index.html.



SS05: Screenshot of To-Do List Web Application

Description: This screenshot shows the deployed React-based To-Do list application. After building and running the React app inside a Docker container, this page confirms the app is functioning as expected and accessible through your VM's IP.



Part 3:

The main challenge I encountered was downloading Ubuntu. Ultimately, I opted to use the Ubuntu Subsystem, conveniently available in the Windows Store, which allowed me to continue with Lab 5. Aside from this initial hurdle, everything else went smoothly.