Running nginx in docker

- Create an EC2 instance (Instance Name: Dockernginx)
- install docker with the following commands:

  sudo yum update

  sudo yum install docker -y

  sudo systemctl start docker

  sudo systemctl status docker (to ensure docker is running)

  sudo chmod 666/var/run.docker.sock

- Pull nginx form docker hub using the following command

  docker pull nginx

```
[ec2-user@ip-192-168-0-47 ~]$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
302e3ee49805: Pull complete
d07412f52e9d: Pull complete
9ab66c386e9c: Pull complete
4b563e5e980a: Pull complete
55af3c8febf2: Pull complete
5b8e768fb22d: Pull complete
85177e2c6f39: Pull complete
Digest: sha256:d2eb56950b84efe34f966a2b92efb1a1a2ea53e7e93b94cdf45a27cf3cd47fc0
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
[ec2-user@ip-192-168-0-47 ~]$
```

- Run a container with name (name :b114vivek_nginx) using the following command
  docker run –name b114vivek_nginx -d -p 8080:80 nginx:latest

```
[ec2-user@ip-192-168-0-47 ~]$ docker run --name b114vivek_nginx -d -p 8080:80 nginx:latest
bcf18d854bf213e3b04a3608de2138617fb0baf962a704564a6656cfed509c60
[ec2-user@ip-192-168-0-47 ~]$
```

- Verify your containers and images in docker using the following commands
  docker ps

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
  NAMES
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp, :::8080->80/tcp
  b114vivek_nginx
```

  docker images

```
[ec2-user@ip-192-168-0-47 ~]$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
nginx         latest    7f553e8bbc89   6 days ago    192MB
[ec2-user@ip-192-168-0-47 ~]$
```

  Here the **container id** of the container created by the **name b114vivek_nginx** is **bcf18d854bf2**

- Now in order to execute the container we have to use the following command

  docker exec -it bcf18d854bf2 bin/bash

```
[ec2-user@ip-192-168-0-47 ~]$ docker exec -it bcf18d854bf2 bin/bash
root@bcf18d854bf2:/#
```

  Now we are inside the container bcf18d854bf2

- Now execute the following commands to get into the html
  cd usr
  cd share
  cd nginx
  cd html

Inside html when using the **ls command** we will find the index.html file is present.

```
[ec2-user@ip-192-168-0-47 ~]$ docker exec -it bcf18d854bf2 bin/bash
root@bcf18d854bf2:/# cd usr
root@bcf18d854bf2:/usr# cd share
root@bcf18d854bf2:/usr/share# cd nginx
root@bcf18d854bf2:/usr/share/nginx# cd html
root@bcf18d854bf2:/usr/share/nginx/html# ls
50x.html   index.html
```

- you can use the index.html file using the command

  cat index.html

```
root@bcf18d854bf2:/usr/share/nginx/html# cat index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@bcf18d854bf2:/usr/share/nginx/html# 
```

- We can edit the index.html using the vim command
  vim index.html

```
root@bcf18d854bf2:/usr/share/nginx/html# vim index.html
bash: vim: command not found
root@bcf18d854bf2:/usr/share/nginx/html# 
```

in case vim command is not found, since we are running the commands inside the container, as container is a light weight weight component and has only essential commands we can install vim command by using the following commands:

apt-get update

```
root@bcf18d854bf2:/usr/share/nginx/html# apt-get update
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8787 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [2468 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [188 kB]
Fetched 9232 kB in 2s (6132 kB/s)
Reading package lists... Done
root@bcf18d854bf2:/usr/share/nginx/html# 
```

[apt-get install vim](#)

```
root@bcf18d854bf2:/usr/share/nginx/html# apt-get install vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 libsodium23 vim-common vim-runtime xxd
Suggested packages:
  gpm ctags vim-doc vim-scripts
The following NEW packages will be installed:
  libgpm2 libsodium23 vim vim-common vim-runtime xxd
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 8976 kB of archives.
After this operation, 41.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main amd64 vim-common all 2:9.0.1378-2 [124 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 libgpm2 amd64 1.20.7-10+b1 [14.2 kB]
```

Since now we have installed vim command we can edit the html as required
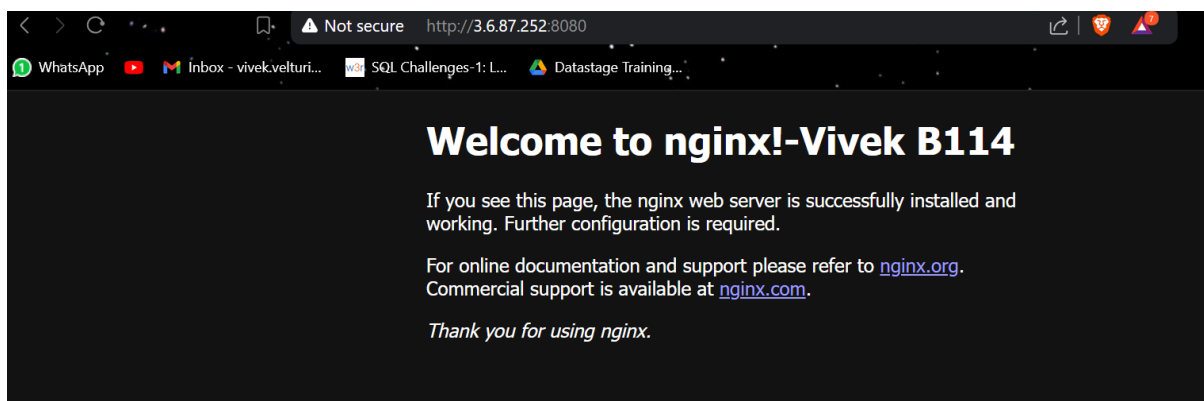
```
root@bcf18d854bf2:/usr/share/nginx/html# vim index.html
root@bcf18d854bf2:/usr/share/nginx/html# cat index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!-Vivek B114</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@bcf18d854bf2:/usr/share/nginx/html# 
```

I have edited the **Welcome to nginx! Output to show Welcome to nginx! -Vivek B114**

- Now up on executing the public ip address of the EC2 with port 8080 we will see the edited html page

http://3.6.87.252:8080

Welcome to nginx!-Vivek B114

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- We can exit the container using the command

  exit

  now we will be out of the container in ec2-user

```
root@bcf18d854bf2:/usr/share/nginx/html# exit
exit
[ec2-user@ip-192-168-0-47 ~]$
```

- If we want we can create a new html file in ec2-user and push it into the container by following the steps listed below:

  Create a new html file and give the required code

```
<!DOCTYPE html>
<html>
<body>

<h1>Greetings: B114</h1>

<p>This is a html code written to push the code into the dcoker container running nginx.</p>

</body>
</html>
~
~
~
~
~
~
~
~
:wq!
```

```
[ec2-user@ip-192-168-0-47 ~]$ vim index.html
[ec2-user@ip-192-168-0-47 ~]$ cat index.html
<!DOCTYPE html>
<html>
<body>

<h1>Greetings: B114</h1>

<p>This is a html code written to push the code into the dcoker container running nginx.</p>

</body>
</html>
[ec2-user@ip-192-168-0-47 ~]$ ls
index.html
[ec2-user@ip-192-168-0-47 ~]$
```

- Since we have the html file outside the container we have to inject inside the container by copying the file to the destination inside the container using the following command

  docker cp index.html bcf18d854bf2:/usr/share/nginx/html

```
[ec2-user@ip-192-168-0-47 ~]$ docker cp index.html bcf18d854bf2:/usr/share/nginx/html
Successfully copied 2.05kB to bcf18d854bf2:/usr/share/nginx/html
[ec2-user@ip-192-168-0-47 ~]$
```

Now upon visiting the public ip address of the ec2 instance we will find the updated html page which we pushed into the container
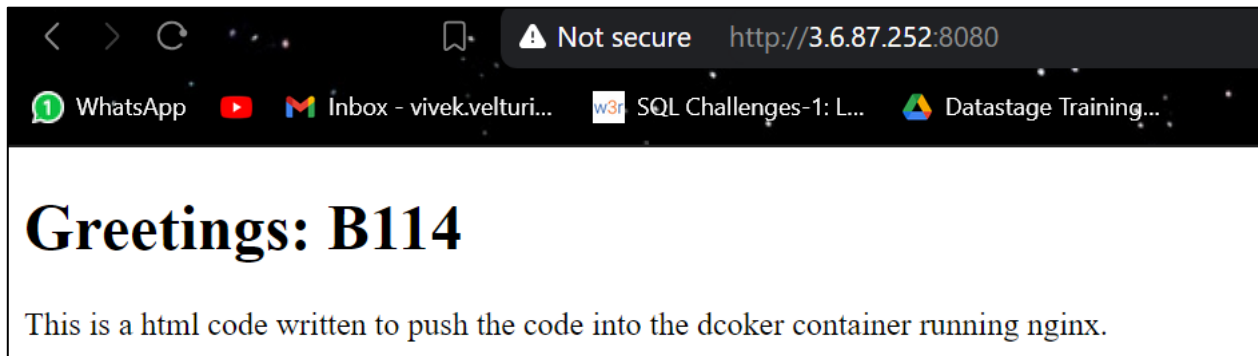
Lets say we want to create another container in the EC2

We can go ahead with the run command as previously given by giving a new container name but we have to make sure the ports are different as same port cannot be allocated to different containers

docker run –name vivek2nginx -d -p 8080:80 nginx:latest

```
[ec2-user@ip-192-168-0-47 ~]$ docker run --name vivek2nginx -d -p 8080:80 nginx:latest
1a7ab16cc975b6434f209cd2caa434c8399b8048baaa1cf3802054fdab6084b5
docker: Error response from daemon: driver failed programming external connectivity on endpoint vivek2nginx (3aca846b2c64ba62ac59f8888
00a770230b68506eabb9b36355f2234ef426725): Bind for 0.0.0.0:8080 failed: port is already allocated.
[ec2-user@ip-192-168-0-47 ~]$ 
```

Container not created as port 8080 is already allocated
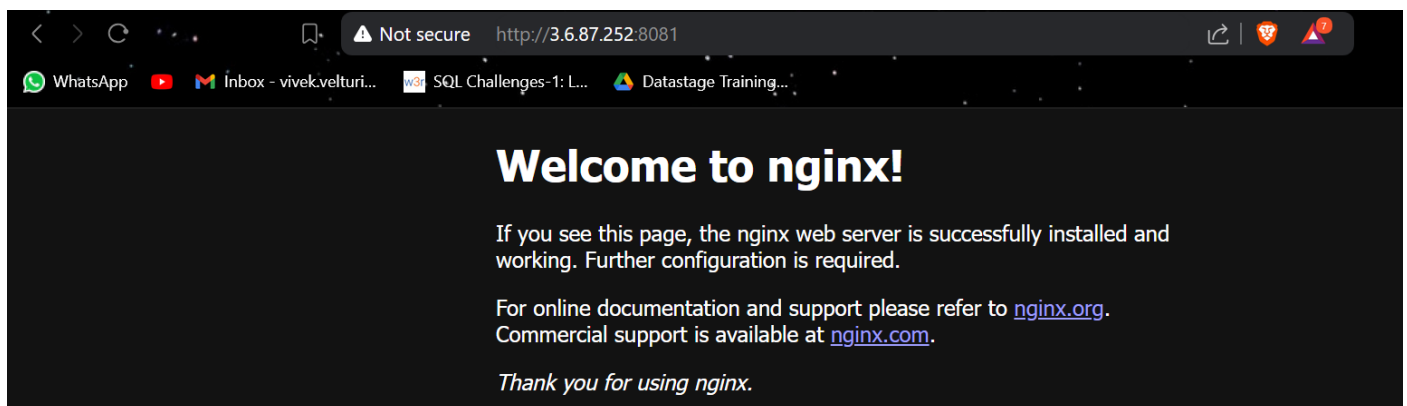
So changing the port and writing the command again

docker run –name vivek2nginx -d -p 8081:80 nginx:latest

```
[ec2-user@ip-192-168-0-47 ~]$ docker run --name vivek2nginx -d -p 8081:80 nginx:latest
61a949e2954911155d463d225dd1557bf532ccea1d3f7f371f40e40699f41110
[ec2-user@ip-192-168-0-47 ~]$ 
```

docker ps command gives you the number of containers now in the EC2

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                                     NAMES
61a949e29549   nginx:latest   "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   0.0.0.0:8081->80/tcp, :::8081->80/tcp    vivek2
nginx
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   46 minutes ago  Up 46 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp    b114vi
vek_nginx
[ec2-user@ip-192-168-0-47 ~]$ 
```

So now upon running the public ip on port 8081 we will get the nginx webpage created in the container vivek2nginx having the container id 61a949e29549

docker ps-aq gives the list of containers available in docker

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps -aq
61a949e29549
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$
```

docker stop <container id> will stop the container mentioned in the command

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps -aq
61a949e29549
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$ docker stop 61a949e29549
61a949e29549
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND            CREATED         STATUS         PORTS                                         NAMES
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   52 minutes ago   Up 52 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   b114vi
vek_nginx
```

docker start <container id> will start the container mentioned in the command

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps -aq
61a949e29549
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$ docker stop 61a949e29549
61a949e29549
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND            CREATED         STATUS         PORTS                                         NAMES
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   52 minutes ago   Up 52 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   b114vi
vek_nginx
[ec2-user@ip-192-168-0-47 ~]$ docker start 61a949e29549
61a949e29549
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE          COMMAND            CREATED         STATUS         PORTS                                         NAMES
61a949e29549   nginx:latest   "/docker-entrypoint.…"   10 minutes ago   Up 2 seconds    0.0.0.0:8081->80/tcp, :::8081->80/tcp   vivek2
nginx
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   53 minutes ago   Up 53 minutes   0.0.0.0:8080->80/tcp, :::8080->80/tcp   b114vi
vek_nginx
[ec2-user@ip-192-168-0-47 ~]$
```

docker rm $ (docker ps -aq) will delete all the containers which are not in running state

```
[ec2-user@ip-192-168-0-47 ~]$ docker rm $( docker ps -aq )
Error response from daemon: cannot remove container "/vivek2nginx": container is running: stop the container before removing or force
remove
Error response from daemon: cannot remove container "/b114vivek_nginx": container is running: stop the container before removing or fo
rce remove
[ec2-user@ip-192-168-0-47 ~]$
```

Since containers are in running state, they are not deleted, so first we have to stop the containers and then execute the command

```
[ec2-user@ip-192-168-0-47 ~]$ docker ps -aq
61a949e29549
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$ docker stop 61a949e29549
61a949e29549
[ec2-user@ip-192-168-0-47 ~]$ docker stop bcf18d854bf2
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$ docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
[ec2-user@ip-192-168-0-47 ~]$ docker ps -a
CONTAINER ID   IMAGE          COMMAND            CREATED         STATUS                  PORTS      NAMES
61a949e29549   nginx:latest   "/docker-entrypoint.…"   14 minutes ago   Exited (0) 33 seconds ago            vivek2nginx
bcf18d854bf2   nginx:latest   "/docker-entrypoint.…"   57 minutes ago   Exited (0) 10 seconds ago            b114vivek_nginx
[ec2-user@ip-192-168-0-47 ~]$ docker rm $( docker ps -aq )
61a949e29549
bcf18d854bf2
[ec2-user@ip-192-168-0-47 ~]$ docker ps -a
CONTAINER ID   IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
[ec2-user@ip-192-168-0-47 ~]$
```