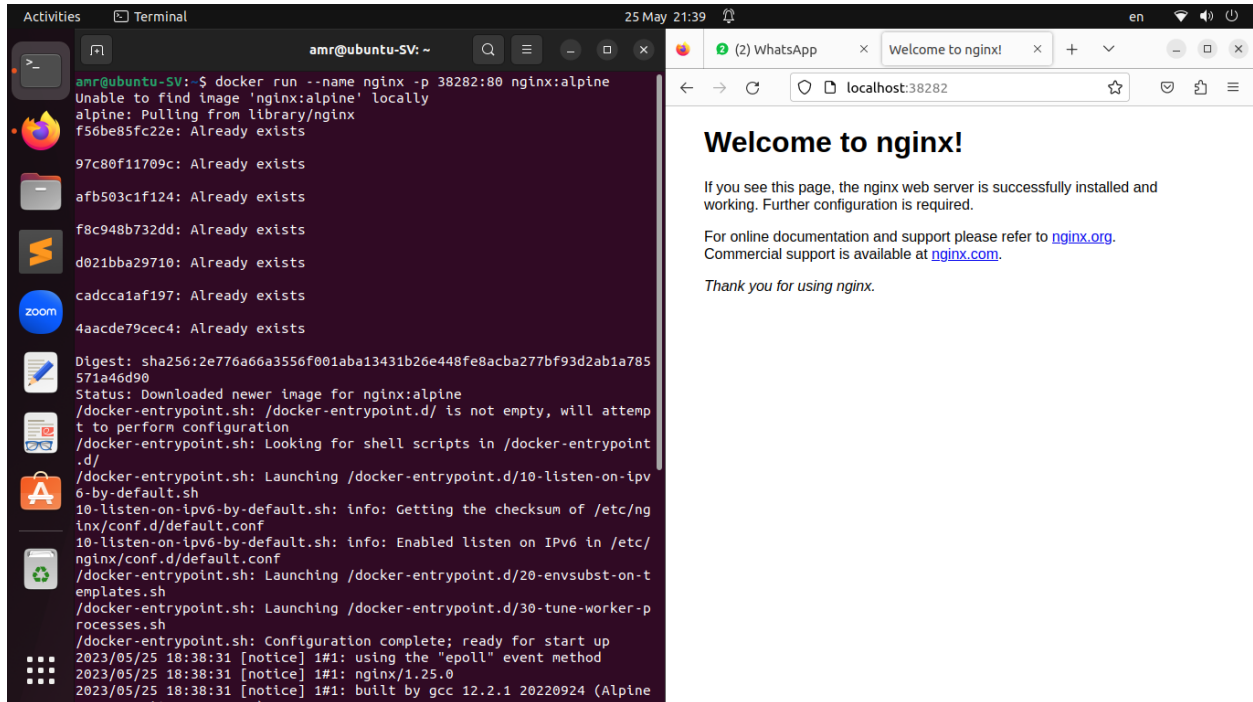


Docker Lab (2)

1-Run an instance of nginx:alpine with a name nginx and map port 80 on the container to 38282 on the host.



The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays the command `docker run --name nginx -p 38282:80 nginx:alpine` and its output, which includes pulling the nginx:alpine image and starting the container. The web browser shows the "Welcome to nginx!" page, indicating that the nginx web server is successfully installed and working.

```
amr@ubuntu-SV: ~$ docker run --name nginx -p 38282:80 nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
f56be85fc22e: Already exists
97c80f11709c: Already exists
afb503c1f124: Already exists
f8c948b732dd: Already exists
d021bba29710: Already exists
cadcca1af197: Already exists
4aacde79cec4: Already exists
Digest: sha256:2e776a66a3556f001aba13431b26e448fe8acba277bf93d2ab1a785571a46d90
Status: Downloaded newer image for nginx:alpine
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/05/25 18:38:31 [notice] 1#1: using the "epoll" event method
2023/05/25 18:38:31 [notice] 1#1: nginx/1.25.0
2023/05/25 18:38:31 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine 12.2.1-r3)

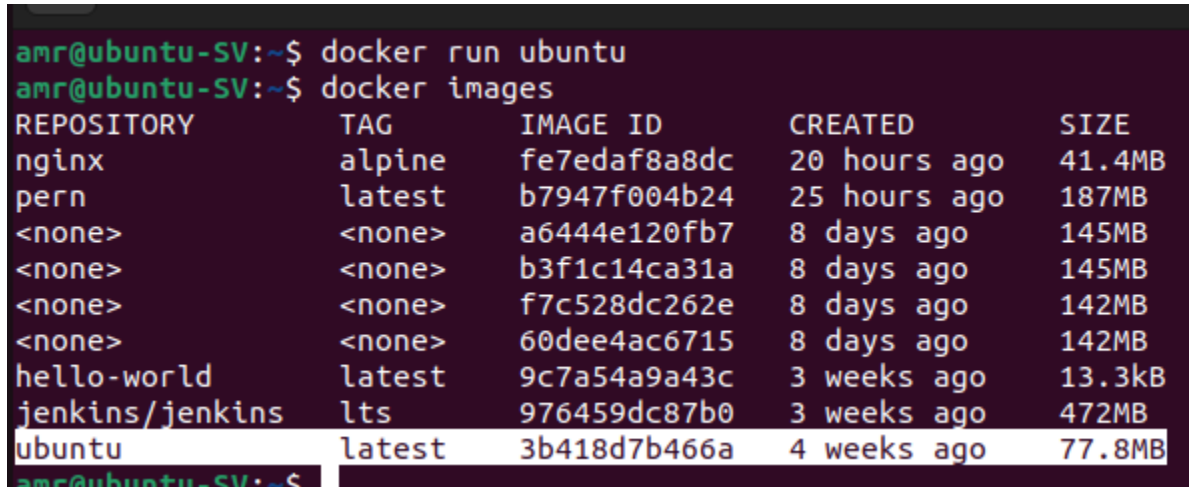
Welcome to nginx!

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.
```

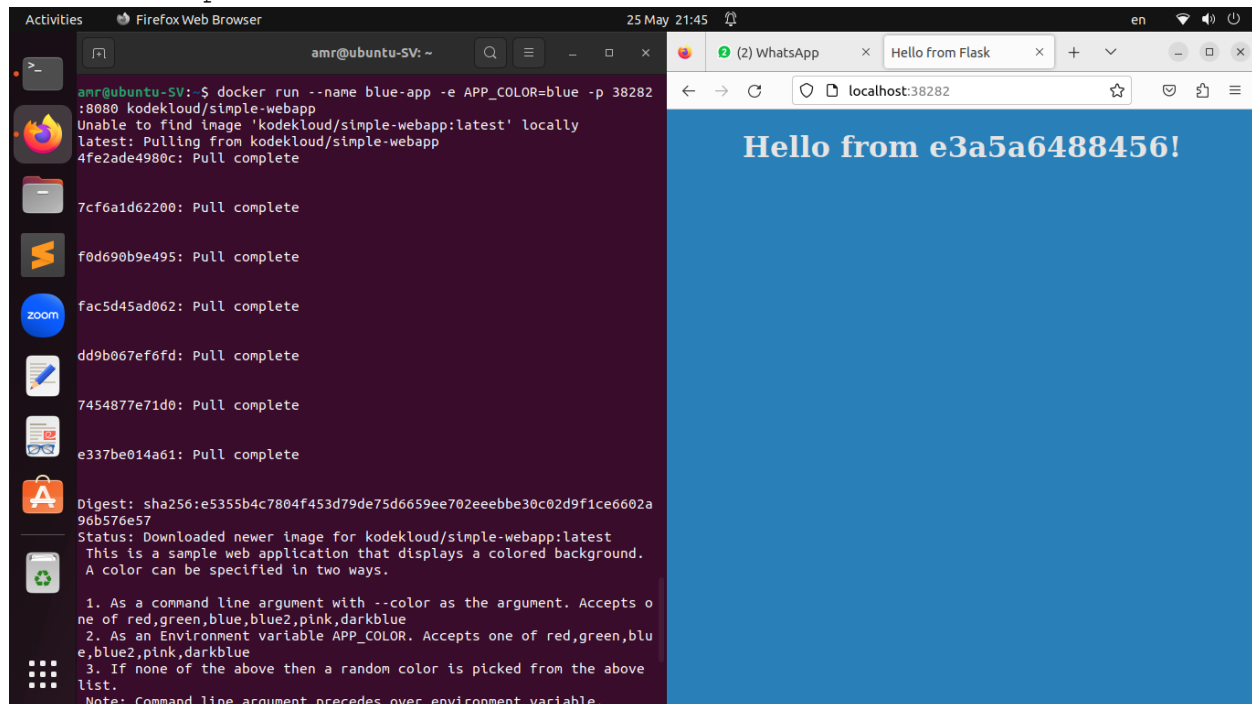
2- create ubuntu image and check the size of it



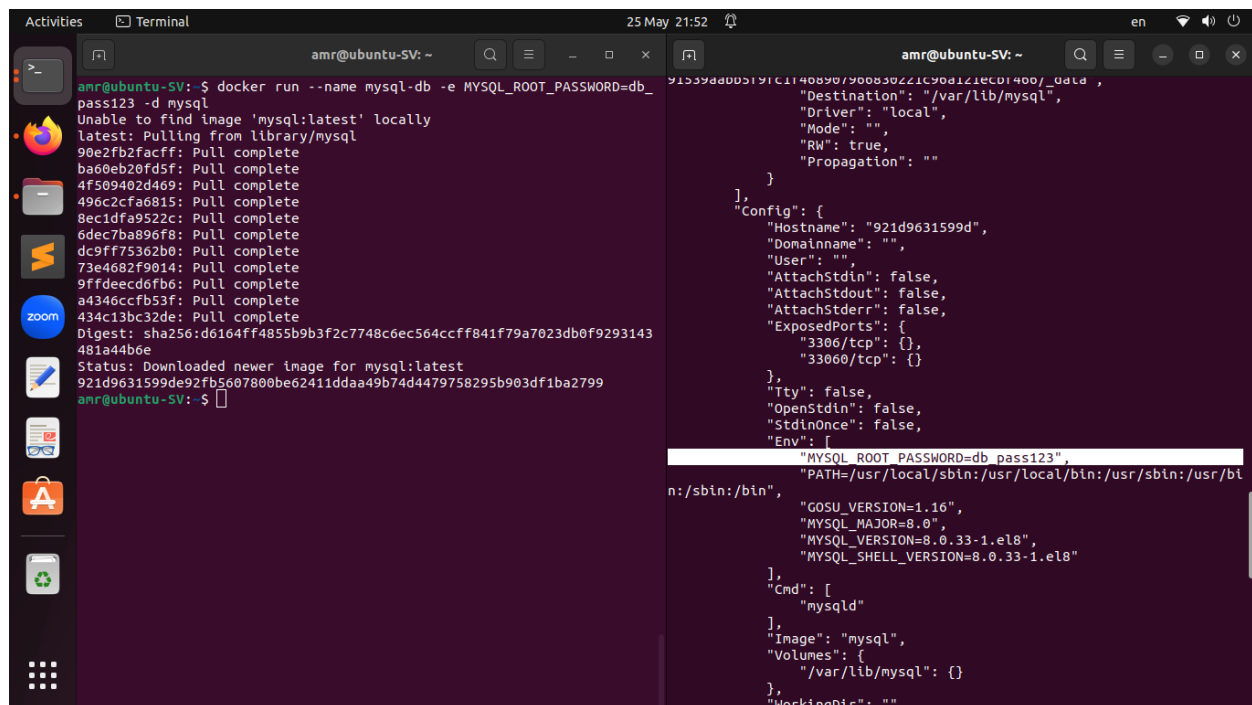
The screenshot shows a terminal window with the command `docker run ubuntu` and the output of the `docker images` command. The output lists the repository, tag, image ID, created time, and size for various images, including nginx:alpine, ubuntu, and others.

```
amr@ubuntu-SV: ~$ docker run ubuntu
amr@ubuntu-SV: ~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                alpine             fe7edaf8a8dc       20 hours ago       41.4MB
pern                 latest             b7947f004b24       25 hours ago       187MB
<none>               <none>             a6444e120fb7       8 days ago         145MB
<none>               <none>             b3f1c14ca31a       8 days ago         145MB
<none>               <none>             f7c528dc262e       8 days ago         142MB
<none>               <none>             60dee4ac6715       8 days ago         142MB
hello-world          latest             9c7a54a9a43c       3 weeks ago        13.3kB
jenkins/jenkins      lts                976459dc87b0       3 weeks ago        472MB
ubuntu               latest             3b418d7b466a       4 weeks ago        77.8MB
amr@ubuntu-SV: ~$
```

3- Run a container named blue-app using image kodekloud/simplewebapp and set the environment variable APP_COLOR to blue. Make the application available on port 38282 on the host. The application listens on port 8080.



4- Deploy a mysql database using the mysql image and name it mysqldb. Set the database password to use db_pass123 then inspect it to check the value



5- pull the code from <https://github.com/sabreensalama/dockerizenode-app-task> and create a docker file for this app

The screenshot shows a terminal window on the right and a web browser on the left. The web browser displays 'Hello World' and a Dockerfile editor. The Dockerfile contains the following content:

```
1 FROM node
2 WORKDIR /app
3 COPY package*.json ./
4 RUN npm install
5 COPY . .
6 EXPOSE 8080
7 CMD ["node", "server.js"]
```

The terminal window shows the command `docker build -t node-app .` and its output, which includes the build progress and the final image ID: `fb3ca87fd89ce6fce2b30ffe285e25fc4c309614953bbd611e02395393783ca`. The terminal also shows the command `docker run -d -p 8080:8080 node-app` and its output, which includes the container ID: `fb3ca87fd89ce6fce2b30ffe285e25fc4c309614953bbd611e02395393783ca`.

6- Create a volume called `mysql_data`, Run a mysql container again, but this time map a volume to the container so that the data stored by the container is stored at `/opt/data` on the host. Use the same name `: mysql-db` and same password: `db_pass123` as before. Mysql stores data at `/var/lib/mysql` inside the container.

The screenshot shows a terminal window with the command `docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql -v /opt/data/mysql_data:/var/lib/mysql mysql` and its output. The output shows the MySQL server starting up and initializing the database files. The output includes the following lines:

```
2023-05-25 19:57:32:00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
2023-05-25 19:57:32:00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-05-25 19:57:33:00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
2023-05-25 19:57:33:00:00 [Note] [Entrypoint]: Initializing database files
2023-05-25T19:57:33.190332Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2023-05-25T19:57:33.251490Z 0 [ERROR] [MY-010083] [Server] --verbose is for use with --help; did you mean --log-error-verbosity?
2023-05-25T19:57:33.251617Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.33) initializing of server in progress as process 80
2023-05-25T19:57:33.310801Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-05-25T19:57:44.059360Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
```