Katie Downing Powell, Branden Solomon, Muhammed Zaid

CTEC 435

Docker Containerization Group Project
https://docs.google.com/document/d/1b_kLylrm36MDAGwAvydP0WSngSobtrP2EKijIATAHRs/edit?usp=sharing

Objective:
Learn to install Docker, manage containers, and run containerized applications on Ubuntu, Linux Mint, Ubuntu Server, and CentOS Server.

Part 1: Installing Docker on Ubuntu 22.04 LTS & Linux Mint 21.3 LTS

1. Update Package List:
```
   sudo apt update
```

2. Install Required Packages:
   Install necessary packages for Docker to function properly.
```
   sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

3. Add Docker GPG Key:
```
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg
```

4. Add Docker Repository:
   Add Docker's official repository to APT sources.
```
   echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

5. Update Package List Again:
```
   sudo apt update
```

6. Install Docker:
```
   sudo apt install docker-ce docker-ce-cli containerd.io
```

7. Start and Enable Docker:
```
```

```
sudo systemctl start docker
sudo systemctl enable docker
```

8. Verify Docker Installation:

```
sudo docker --version
```

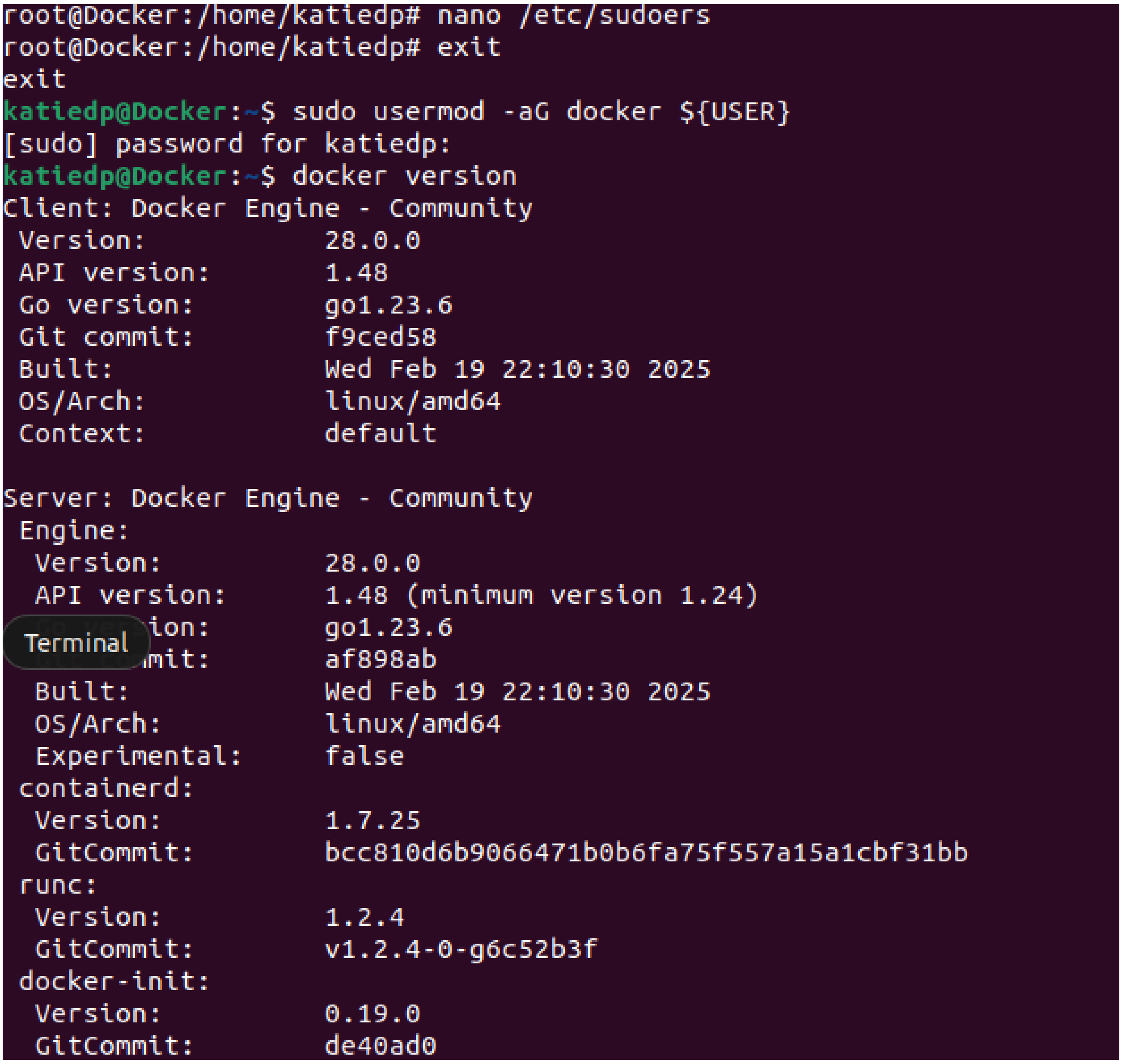
- It should return the installed version of Docker.

9. (Optional) Run Docker as Non-Root User:

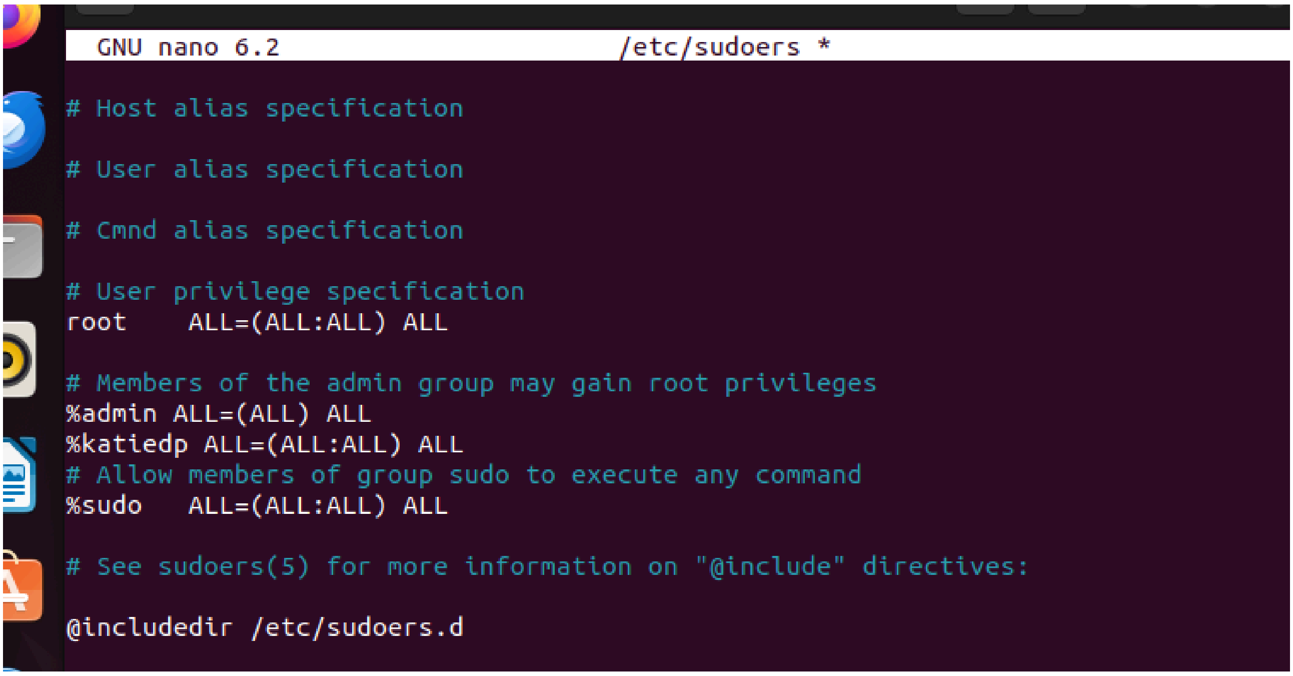
Add your user to the `docker` group to run Docker commands without `sudo`.

```
sudo usermod -aG docker ${USER}
```

Log out and back in for this change to take effect.



```
root@Docker:/home/katiedp# nano /etc/sudoers
root@Docker:/home/katiedp# exit
exit
katiedp@Docker:~$ sudo usermod -aG docker ${USER}
[sudo] password for katiedp:
katiedp@Docker:~$ docker version
Client: Docker Engine - Community
 Version:           28.0.0
 API version:       1.48
 Go version:        go1.23.6
 Git commit:        f9ced58
 Built:             Wed Feb 19 22:10:30 2025
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          28.0.0
  API version:      1.48 (minimum version 1.24)
  Go version:       go1.23.6
  Git commit:       af898ab
  Built:            Wed Feb 19 22:10:30 2025
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.7.25
  GitCommit:        bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
 runc:
  Version:          1.2.4
  GitCommit:        v1.2.4-0-g6c52b3f
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```



```
  GNU nano 6.2                            /etc/sudoers *

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
%katiedp ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```

---Part 1 is Complete 2/22/2025

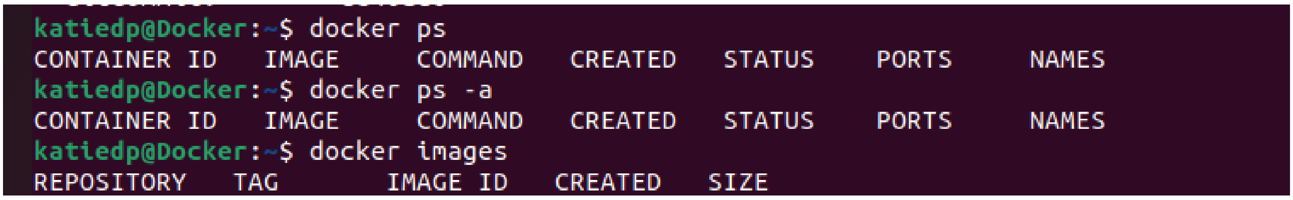
Part 2: Managing Docker Containers and Images—-Start here with group

Task 1: Listing Containers and Images

1. List Running Containers:

   - Use `docker ps` to list all running containers:

```
docker ps
```

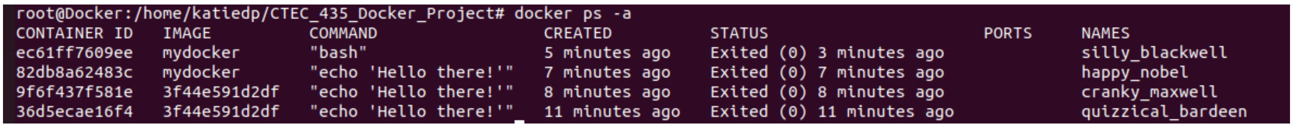


```
katiedp@Docker:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
katiedp@Docker:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
katiedp@Docker:~$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

2. List All Containers (Including Stopped Containers):

   - Use `docker ps -a` to list all containers, both running and stopped:

```
docker ps -a
```

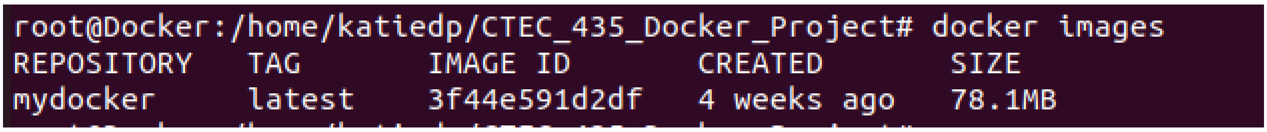


```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                       PORTS     NAMES
ec61ff7609ee   mydocker       "bash"                   5 minutes ago    Exited (0) 3 minutes ago               silly_blackwell
82db8a62483c   mydocker       "echo 'Hello there!'"    7 minutes ago    Exited (0) 7 minutes ago               happy_nobel
9f6f437f581e   3f44e591d2df   "echo 'Hello there!'"    8 minutes ago    Exited (0) 8 minutes ago               cranky_maxwell
36d5ecae16f4   3f44e591d2df   "echo 'Hello there!'"    11 minutes ago   Exited (0) 11 minutes ago              quizzical_bardeen
```

3. List Docker Images:

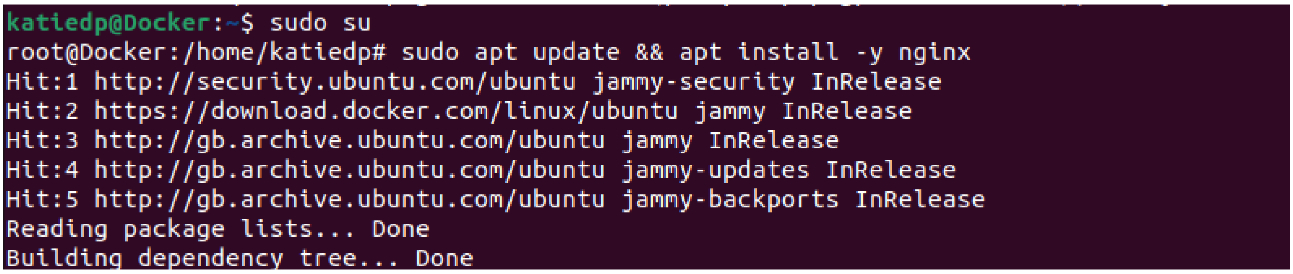
   - Use `docker images` to list all downloaded and built Docker images:

```
docker images
```

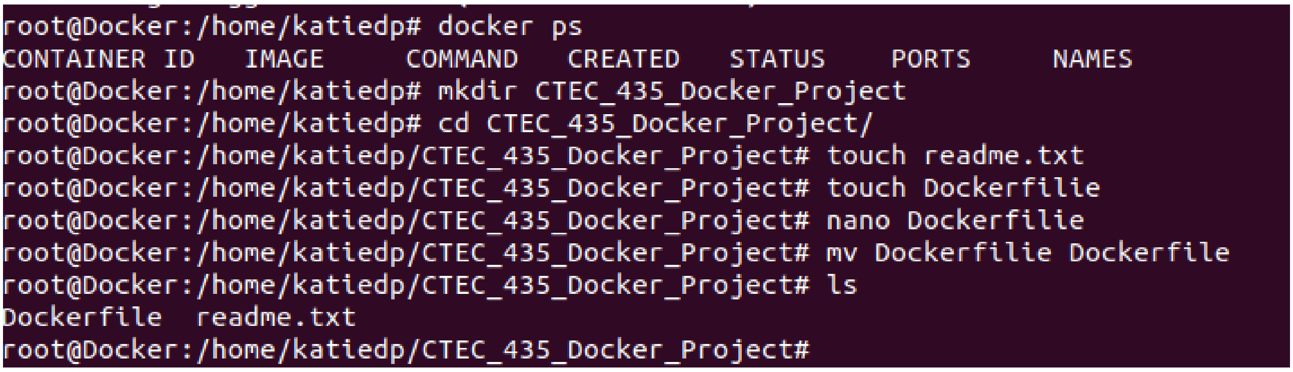


```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mydocker      latest    3f44e591d2df   4 weeks ago    78.1MB
```

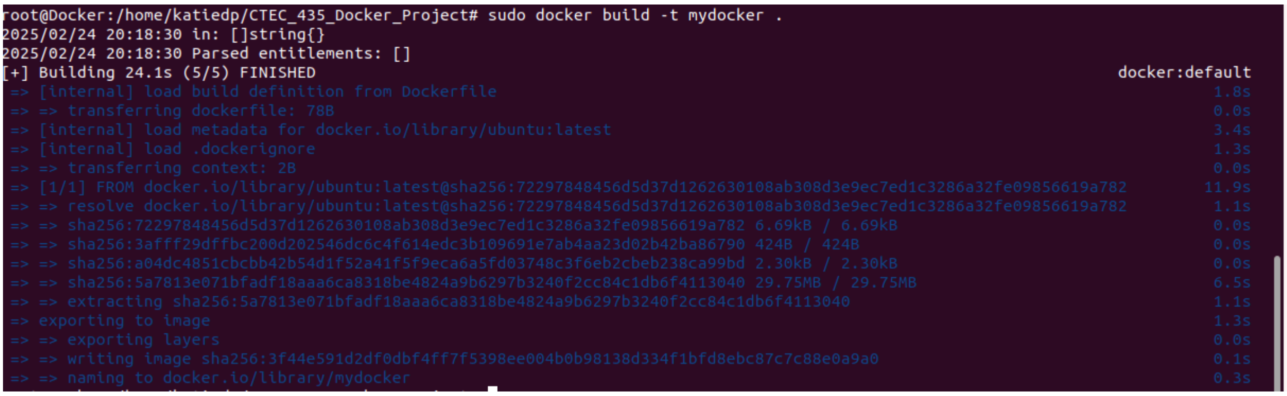
Installing Docker Containers



```
katiedp@Docker:~$ sudo su
root@Docker:/home/katiedp# sudo apt update && apt install -y nginx
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:3 http://gb.archive.ubuntu.com/ubuntu jammy InRelease
Hit:4 http://gb.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:5 http://gb.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
```

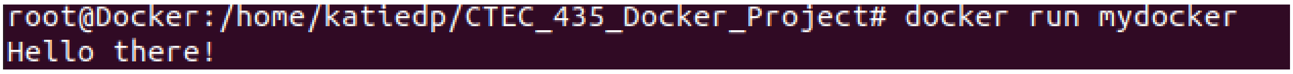


```
root@Docker:/home/katiedp# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@Docker:/home/katiedp# mkdir CTEC_435_Docker_Project
root@Docker:/home/katiedp# cd CTEC_435_Docker_Project/
root@Docker:/home/katiedp/CTEC_435_Docker_Project# touch readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project# touch Dockerfilie
root@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfilie
root@Docker:/home/katiedp/CTEC_435_Docker_Project# mv Dockerfilie Dockerfile
root@Docker:/home/katiedp/CTEC_435_Docker_Project# ls
Dockerfile  readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

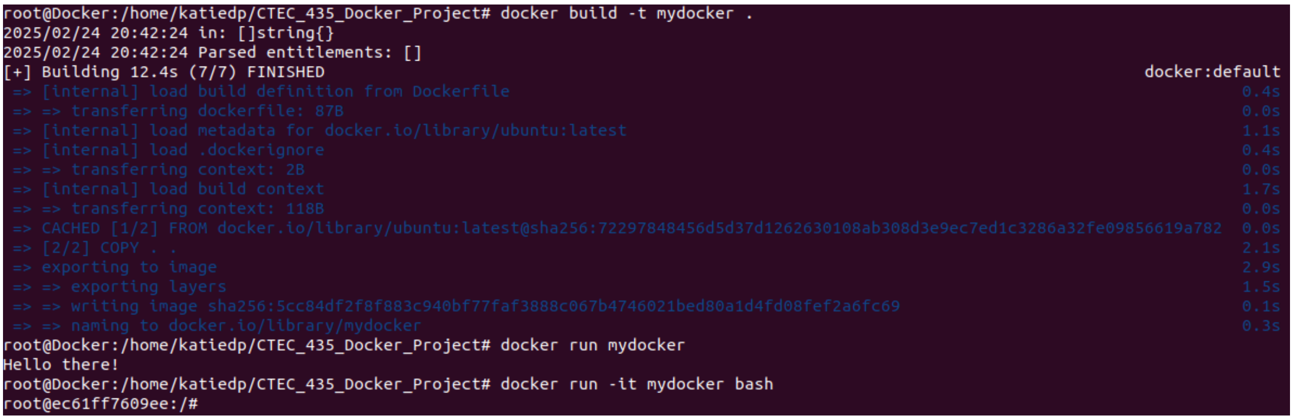


```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# sudo docker build -t mydocker .
2025/02/24 20:18:30 in: []string{}
2025/02/24 20:18:30 Parsed entitlements: []
[+] Building 24.1s (5/5) FINISHED                                                                           docker:default
 => [internal] load build definition from Dockerfile                                                                  1.8s
 => => transferring dockerfile: 78B                                                                                   0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                      3.4s
 => [internal] load .dockerignore                                                                                     1.3s
 => => transferring context: 2B                                                                                       0.0s
 => [1/1] FROM docker.io/library/ubuntu:latest@sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782      11.9s
 => => resolve docker.io/library/ubuntu:latest@sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782       1.1s
 => => sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782 6.69kB / 6.69kB                        0.0s
 => => sha256:3afff29dffbc200d202546dc6c4f614edc3b109691e7ab4aa23d02b42ba86790 424B / 424B                            0.0s
 => => sha256:a04dc4851cbcbb42b54d1f52a41f5f9eca6a5fd03748c3f6eb2cbeb238ca99bd 2.30kB / 2.30kB                        0.0s
 => => sha256:5a7813e071bfadf18aaa6ca8318be4824a9b6297b3240f2cc84c1db6f4113040 29.75MB / 29.75MB                      6.5s
 => => extracting sha256:5a7813e071bfadf18aaa6ca8318be4824a9b6297b3240f2cc84c1db6f4113040                             1.1s
 => exporting to image                                                                                                1.3s
 => => exporting layers                                                                                               0.0s
 => => writing image sha256:3f44e591d2df0dbf4ff7f5398ee004b0b98138d334f1bfd8ebc87c7c88e0a9a0                          0.1s
 => => naming to docker.io/library/mydocker                                                                           0.3s
```

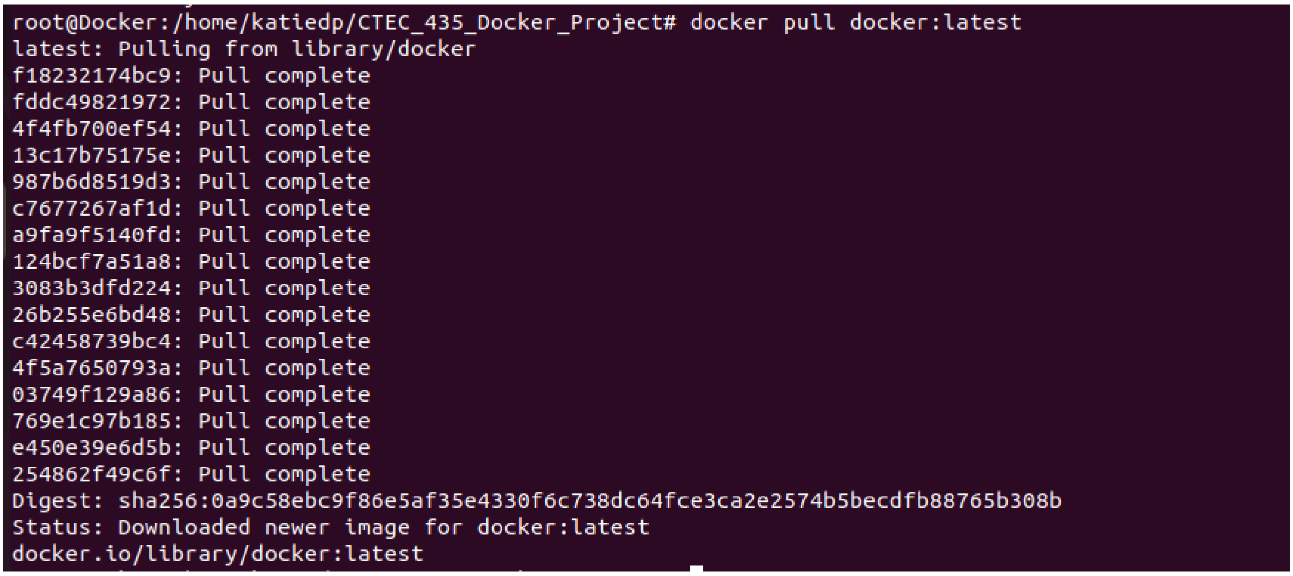
Run Docker Image



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run mydocker
Hello there!
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t mydocker .
2025/02/24 20:42:24 in: []string{}
2025/02/24 20:42:24 Parsed entitlements: []
[+] Building 12.4s (7/7) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                           0.4s
 => => transferring dockerfile: 87B                                                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                               1.1s
 => [internal] load .dockerignore                                                                                              0.4s
 => => transferring context: 2B                                                                                                0.0s
 => [internal] load build context                                                                                              1.7s
 => => transferring context: 118B                                                                                              0.0s
 => CACHED [1/2] FROM docker.io/library/ubuntu:latest@sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782  0.0s
 => [2/2] COPY . .                                                                                                             2.1s
 => exporting to image                                                                                                         2.9s
 => => exporting layers                                                                                                        1.5s
 => => writing image sha256:5cc84df2f8f883c940bf77faf3888c067b4746021bed80a1d4fd08fef2a6fc69                                   0.1s
 => => naming to docker.io/library/mydocker                                                                                    0.3s
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run mydocker
Hello there!
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -it mydocker bash
root@ec61ff7609ee:/#
```



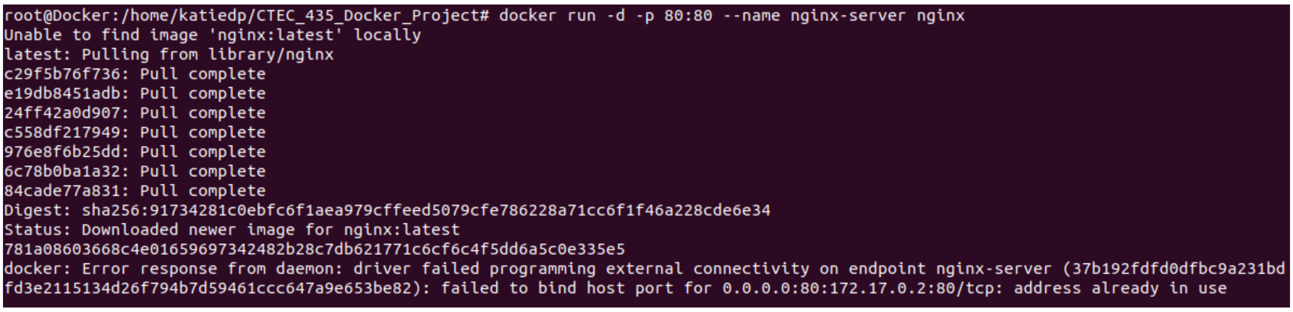
```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker pull docker:latest
latest: Pulling from library/docker
f18232174bc9: Pull complete
fddc49821972: Pull complete
4f4fb700ef54: Pull complete
13c17b75175e: Pull complete
987b6d8519d3: Pull complete
c7677267af1d: Pull complete
a9fa9f5140fd: Pull complete
124bcf7a51a8: Pull complete
3083b3dfd224: Pull complete
26b255e6bd48: Pull complete
c42458739bc4: Pull complete
4f5a7650793a: Pull complete
03749f129a86: Pull complete
769e1c97b185: Pull complete
e450e39e6d5b: Pull complete
254862f49c6f: Pull complete
Digest: sha256:0a9c58ebc9f86e5af35e4330f6c738dc64fce3ca2e2574b5becdfb88765b308b
Status: Downloaded newer image for docker:latest
docker.io/library/docker:latest
```

Task 2: Starting and Managing a Basic Container

1. Run a Nginx Container:

- Start an Nginx web server container:

  ```

  docker run -d -p 80:80 --name nginx-server nginx
  ```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -d -p 80:80 --name nginx-server nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
c29f5b76f736: Pull complete
e19db8451adb: Pull complete
24ff42a0d907: Pull complete
c558df217949: Pull complete
976e8f6b25dd: Pull complete
6c78b0ba1a32: Pull complete
84cade77a831: Pull complete
Digest: sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Status: Downloaded newer image for nginx:latest
781a08603668c4e01659697342482b28c7db621771c6cf6c4f5dd6a5c0e335e5
docker: Error response from daemon: driver failed programming external connectivity on endpoint nginx-server (37b192fdfd0dfbc9a231bd
fd3e2115134d26f794b7d59461ccc647a9e653be82): failed to bind host port for 0.0.0.0:80:172.17.0.2:80/tcp: address already in use
```

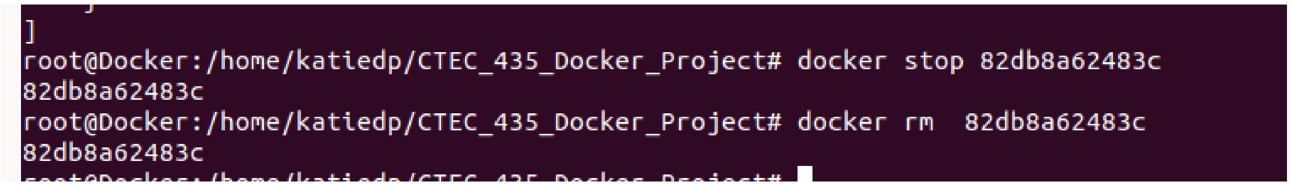
2. Check Running Containers:
   - Verify that the Nginx container is running using `docker ps`.

3. Access the Web Server:
   - Open your browser and navigate to your server's IP to access the default Nginx page.

Task 3: Removing Containers and Images

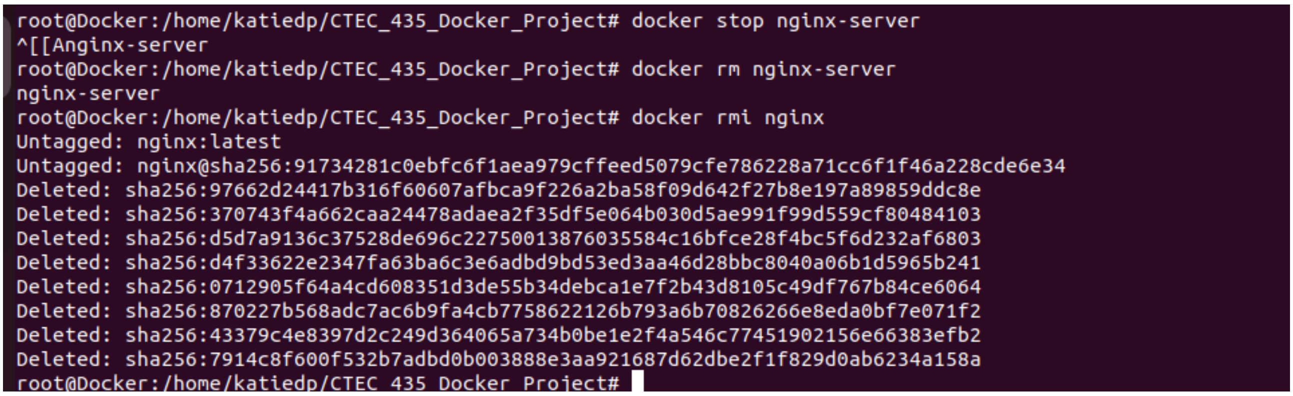
1. Stop and Remove a Container:
   - Stop and remove the running Nginx container:

```
docker stop nginx-server
docker rm nginx-server
```



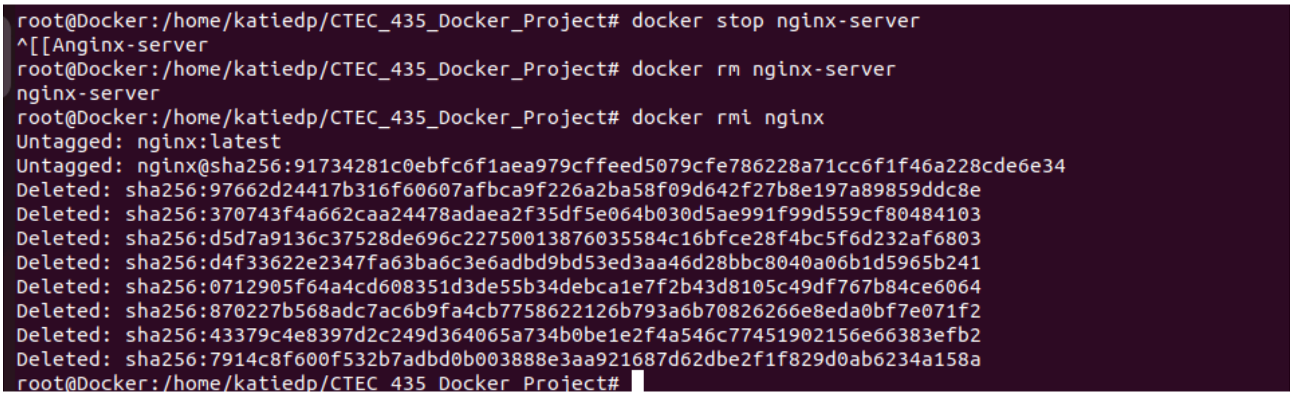
```
]
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker stop 82db8a62483c
82db8a62483c
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker rm  82db8a62483c
82db8a62483c
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

2. Remove an Unused Image:
   - Remove the Nginx image:



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker stop nginx-server
^[[Anginx-server
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker rm nginx-server
nginx-server
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Deleted: sha256:97662d24417b316f60607afbca9f226a2ba58f09d642f27b8e197a89859ddc8e
Deleted: sha256:370743f4a662caa24478adaea2f35df5e064b030d5ae991f99d559cf80484103
Deleted: sha256:d5d7a9136c37528de696c22750013876035584c16bfce28f4bc5f6d232af6803
Deleted: sha256:d4f33622e2347fa63ba6c3e6adbd9bd53ed3aa46d28bbc8040a06b1d5965b241
Deleted: sha256:0712905f64a4cd608351d3de55b34debca1e7f2b43d8105c49df767b84ce6064
Deleted: sha256:870227b568adc7ac6b9fa4cb7758622126b793a6b70826266e8eda0bf7e071f2
Deleted: sha256:43379c4e8397d2c249d364065a734b0be1e2f4a546c77451902156e66383efb2
Deleted: sha256:7914c8f600f532b7adbd0b003888e3aa921687d62dbe2f1f829d0ab6234a158a
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

```
docker rmi nginx
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker stop nginx-server
^[[Anginx-server
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker rm nginx-server
nginx-server
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker rmi nginx
Untagged: nginx:latest
Untagged: nginx@sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34
Deleted: sha256:97662d24417b316f60607afbca9f226a2ba58f09d642f27b8e197a89859ddc8e
Deleted: sha256:370743f4a662caa24478adaea2f35df5e064b030d5ae991f99d559cf80484103
Deleted: sha256:d5d7a9136c37528de696c22750013876035584c16bfce28f4bc5f6d232af6803
Deleted: sha256:d4f33622e2347fa63ba6c3e6adbd9bd53ed3aa46d28bbc8040a06b1d5965b241
Deleted: sha256:0712905f64a4cd608351d3de55b34debca1e7f2b43d8105c49df767b84ce6064
Deleted: sha256:870227b568adc7ac6b9fa4cb7758622126b793a6b70826266e8eda0bf7e071f2
Deleted: sha256:43379c4e8397d2c249d364065a734b0be1e2f4a546c77451902156e66383efb2
Deleted: sha256:7914c8f600f532b7adbd0b003888e3aa921687d62dbe2f1f829d0ab6234a158a
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

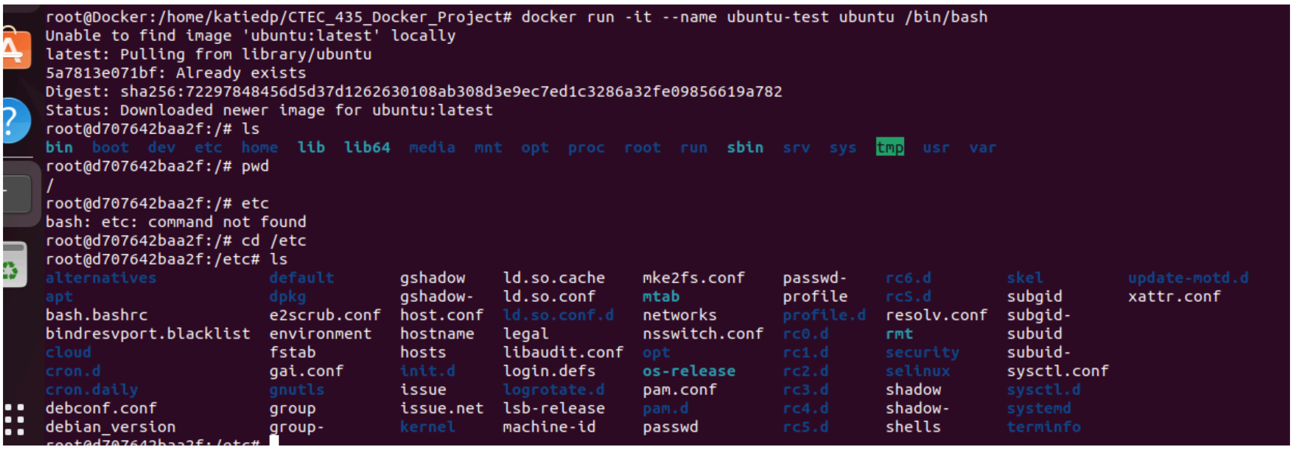
Part 3: Interacting with Docker Containers

Task 4: Getting Shell Access to a Running Container

1. Run a Container with Interactive Shell Access:

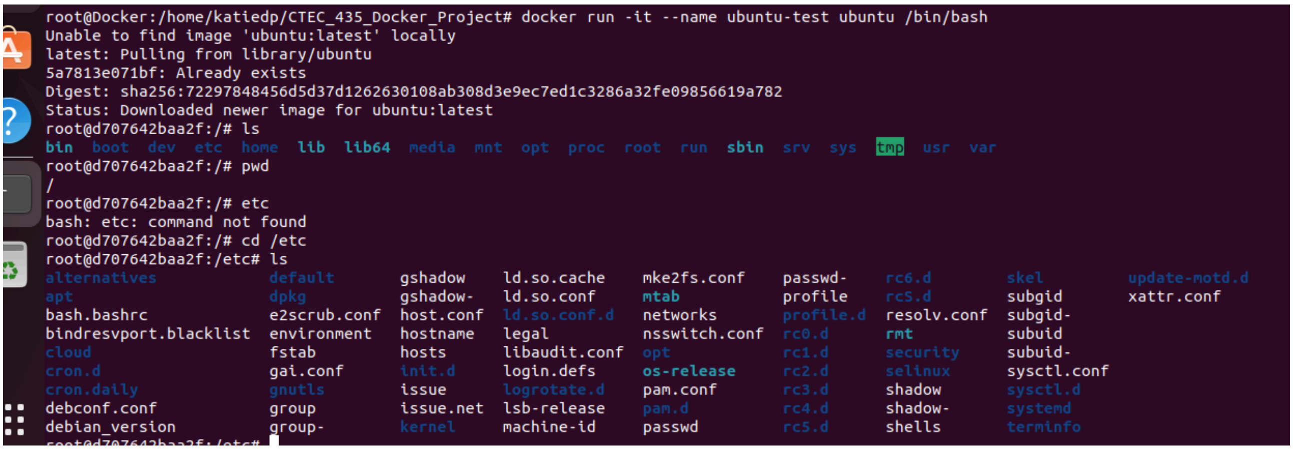
- Run a new container and get shell access inside it:

```
docker run -it --name ubuntu-test ubuntu /bin/bash
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -it --name ubuntu-test ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5a7813e071bf: Already exists
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
root@d707642baa2f:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@d707642baa2f:/# pwd
/
root@d707642baa2f:/# etc
bash: etc: command not found
root@d707642baa2f:/# cd /etc
root@d707642baa2f:/etc# ls
alternatives            default       gshadow     ld.so.cache     mke2fs.conf    passwd-    rc6.d         skel         update-motd.d
apt                     dpkg          gshadow-    ld.so.conf      mtab           profile    rcS.d         subgid       xattr.conf
bash.bashrc             e2scrub.conf  host.conf   ld.so.conf.d    networks       profile.d  resolv.conf   subgid-
bindresvport.blacklist  environment   hostname    legal           nsswitch.conf  rc0.d      rmt           subuid
cloud                   fstab         hosts       libaudit.conf   opt            rc1.d      security      subuid-
cron.d                  gai.conf      init.d      login.defs      os-release     rc2.d      selinux       sysctl.conf
cron.daily              gnutls        issue       logrotate.d     pam.conf       rc3.d      shadow        sysctl.d
debconf.conf            group         issue.net   lsb-release     pam.d          rc4.d      shadow-       systemd
debian_version          group-        kernel      machine-id      passwd         rc5.d      shells        terminfo
root@d707642baa2f:/etc#
```

- Explore the container’s file system and environment by running typical Linux commands (`ls`, `pwd`, etc.).



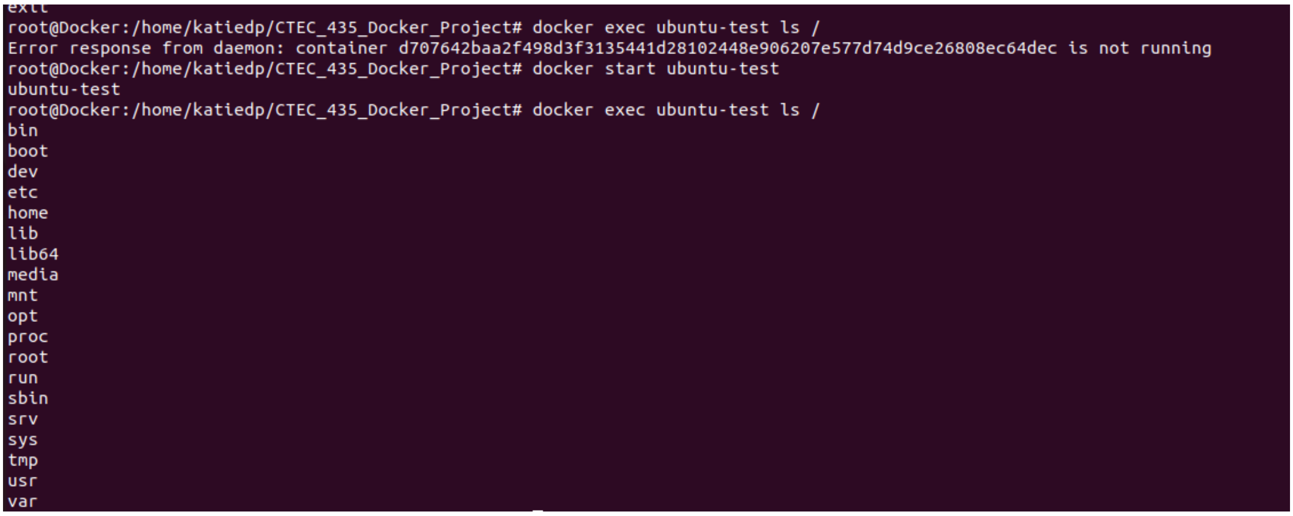
```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -it --name ubuntu-test ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
5a7813e071bf: Already exists
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
root@d707642baa2f:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@d707642baa2f:/# pwd
/
root@d707642baa2f:/# etc
bash: etc: command not found
root@d707642baa2f:/# cd /etc
root@d707642baa2f:/etc# ls
alternatives            default       gshadow     ld.so.cache     mke2fs.conf    passwd-    rc6.d         skel         update-motd.d
apt                     dpkg          gshadow-    ld.so.conf      mtab           profile    rcS.d         subgid       xattr.conf
bash.bashrc             e2scrub.conf  host.conf   ld.so.conf.d    networks       profile.d  resolv.conf   subgid-
bindresvport.blacklist  environment   hostname    legal           nsswitch.conf  rc0.d      rmt           subuid
cloud                   fstab         hosts       libaudit.conf   opt            rc1.d      security      subuid-
cron.d                  gai.conf      init.d      login.defs      os-release     rc2.d      selinux       sysctl.conf
cron.daily              gnutls        issue       logrotate.d     pam.conf       rc3.d      shadow        sysctl.d
debconf.conf            group         issue.net   lsb-release     pam.d          rc4.d      shadow-       systemd
debian_version          group-        kernel      machine-id      passwd         rc5.d      shells        terminfo
root@d707642baa2f:/etc#
```

2. Exit the Container:
   - Exit the shell using the `exit` command.

Task 5: Executing Commands Inside a Running Container

1. Execute a Command Inside a Running Container:
   - Run a command inside the `ubuntu-test` container without entering the shell:

```
docker exec ubuntu-test ls /
```

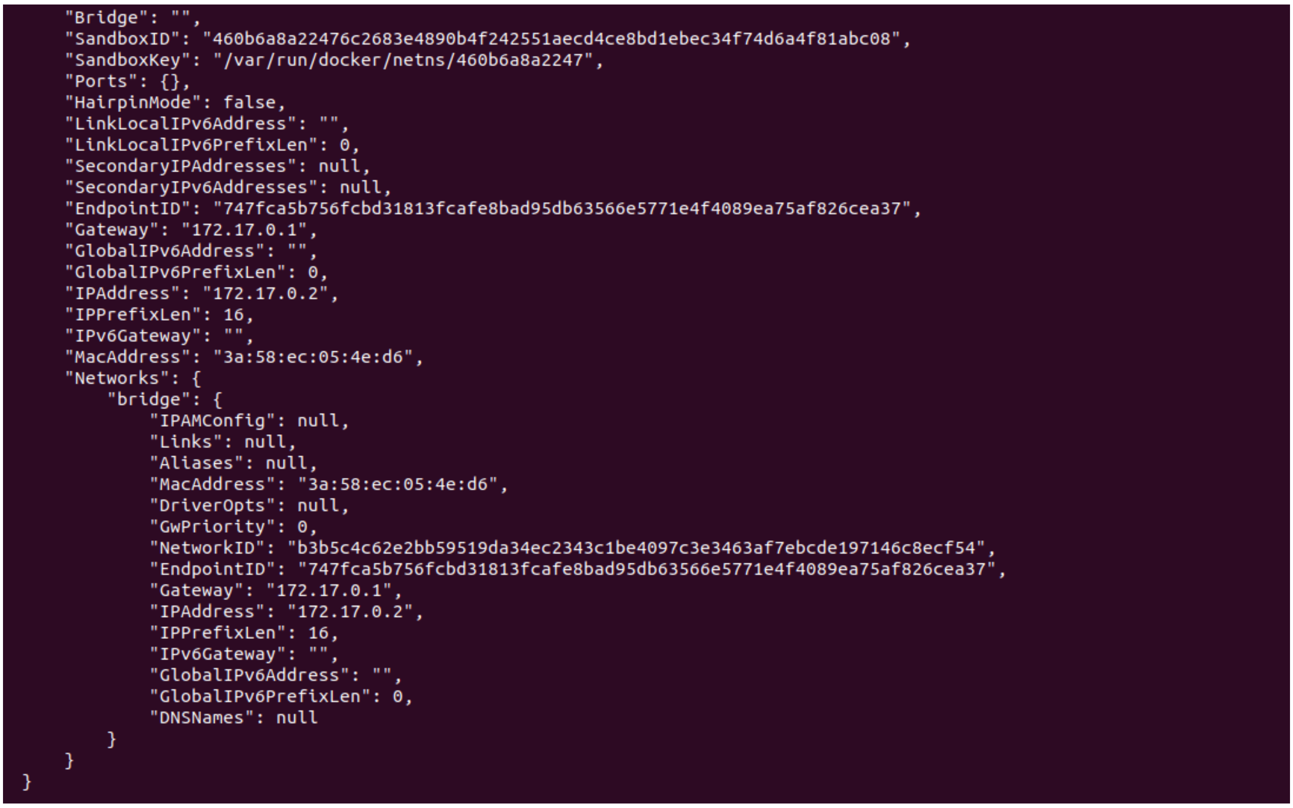


```
exit
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec ubuntu-test ls /
Error response from daemon: container d707642baa2f498d3f3135441d28102448e906207e577d74d9ce26808ec64dec is not running
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker start ubuntu-test
ubuntu-test
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec ubuntu-test ls /
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

Task 6: Inspecting a Container's Information

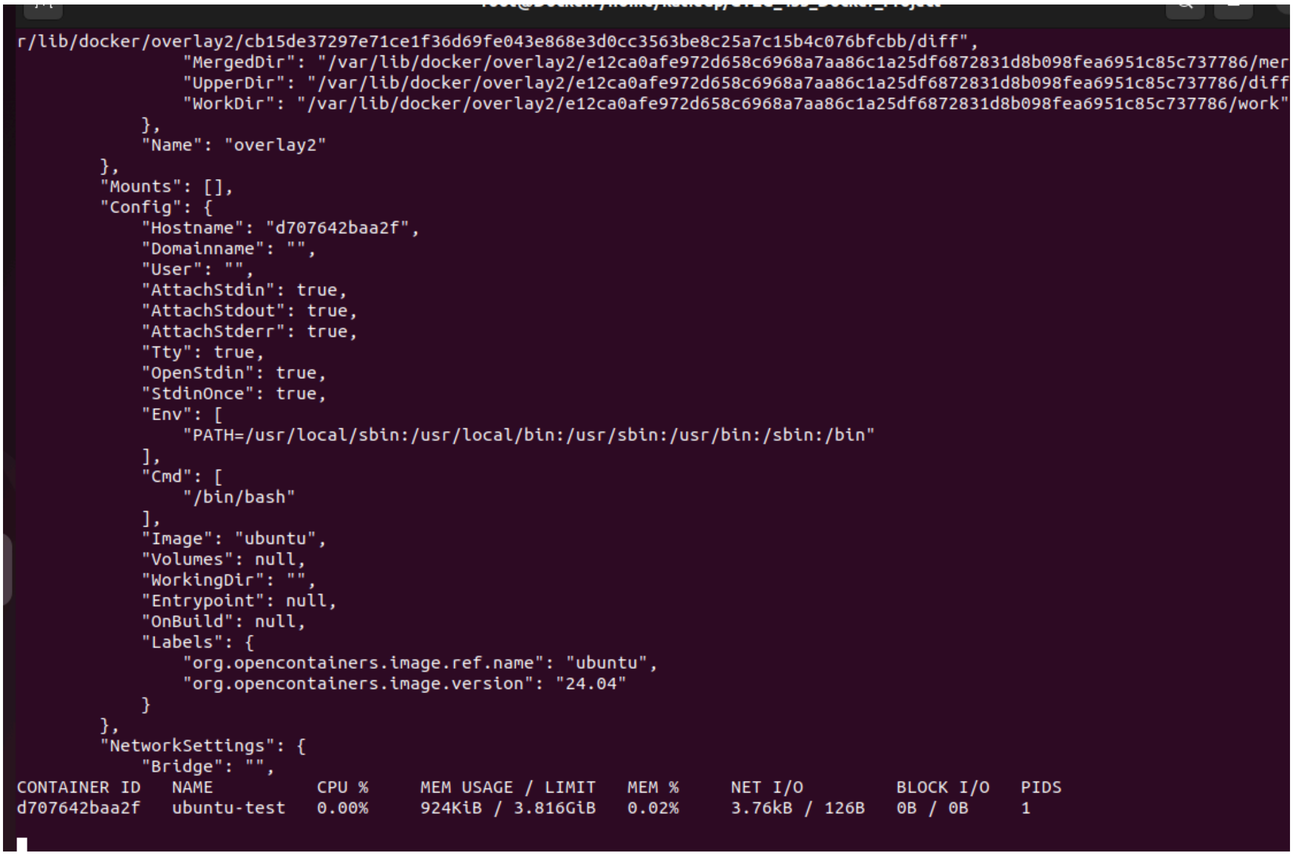
1. Inspect Container Configuration:
   - Get detailed information about the running container with the `docker inspect` command:

```
docker inspect ubuntu-test
```



~~~
        "Bridge": "",
        "SandboxID": "460b6a8a22476c2683e4890b4f242551aecd4ce8bd1ebec34f74d6a4f81abc08",
        "SandboxKey": "/var/run/docker/netns/460b6a8a2247",
        "Ports": {},
        "HairpinMode": false,
        "LinkLocalIPv6Address": "",
        "LinkLocalIPv6PrefixLen": 0,
        "SecondaryIPAddresses": null,
        "SecondaryIPv6Addresses": null,
        "EndpointID": "747fca5b756fcbd31813fcafe8bad95db63566e5771e4f4089ea75af826cea37",
        "Gateway": "172.17.0.1",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "MacAddress": "3a:58:ec:05:4e:d6",
        "Networks": {
            "bridge": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": null,
                "MacAddress": "3a:58:ec:05:4e:d6",
                "DriverOpts": null,
                "GwPriority": 0,
                "NetworkID": "b3b5c4c62e2bb59519da34ec2343c1be4097c3e3463af7ebcde197146c8ecf54",
                "EndpointID": "747fca5b756fcbd31813fcafe8bad95db63566e5771e4f4089ea75af826cea37",
                "Gateway": "172.17.0.1",
                "IPAddress": "172.17.0.2",
                "IPPrefixLen": 16,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DNSNames": null
            }
        }
    }
~~~

```

2. View Resource Usage:
   - Use the `docker stats` command to view resource usage (CPU, memory, etc.) for the running containers:

```
docker stats



```
r/lib/docker/overlay2/cb15de37297e71ce1f36d69fe043e868e3d0cc3563be8c25a7c15b4c076bfcbb/diff",
                "MergedDir": "/var/lib/docker/overlay2/e12ca0afe972d658c6968a7aa86c1a25df6872831d8b098fea6951c85c737786/mer
                "UpperDir": "/var/lib/docker/overlay2/e12ca0afe972d658c6968a7aa86c1a25df6872831d8b098fea6951c85c737786/diff
                "WorkDir": "/var/lib/docker/overlay2/e12ca0afe972d658c6968a7aa86c1a25df6872831d8b098fea6951c85c737786/work"
            },
            "Name": "overlay2"
        },
        "Mounts": [],
        "Config": {
            "Hostname": "d707642baa2f",
            "Domainname": "",
            "User": "",
            "AttachStdin": true,
            "AttachStdout": true,
            "AttachStderr": true,
            "Tty": true,
            "OpenStdin": true,
            "StdinOnce": true,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/bash"
            ],
            "Image": "ubuntu",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": {
                "org.opencontainers.image.ref.name": "ubuntu",
                "org.opencontainers.image.version": "24.04"
            }
        },
        "NetworkSettings": {
            "Bridge": "",
CONTAINER ID   NAME          CPU %     MEM USAGE / LIMIT     MEM %     NET I/O          BLOCK I/O    PIDS
d707642baa2f   ubuntu-test   0.00%     924KiB / 3.816GiB     0.02%     3.76kB / 126B    0B / 0B      1
```

\`\`\`

Part 4: Docker Images, Customization, and Docker Hub

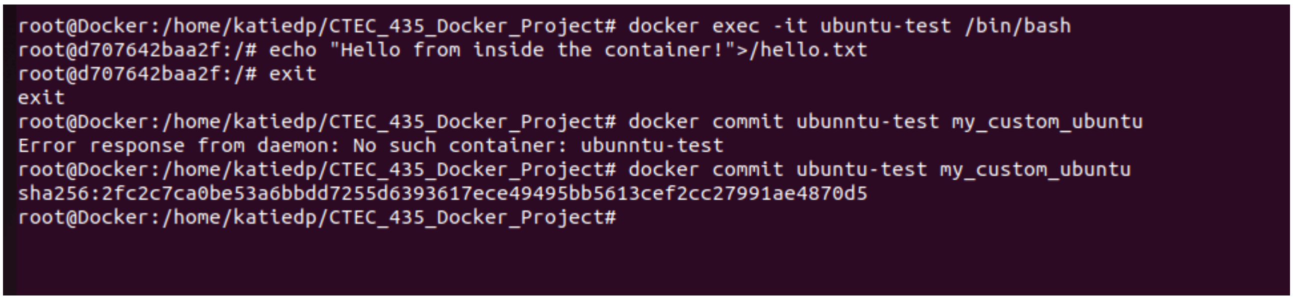
Task 7: Committing Container Changes to a New Image

1. Make Changes Inside a Container:

- Enter the `ubuntu-test` container:

\`\`\`

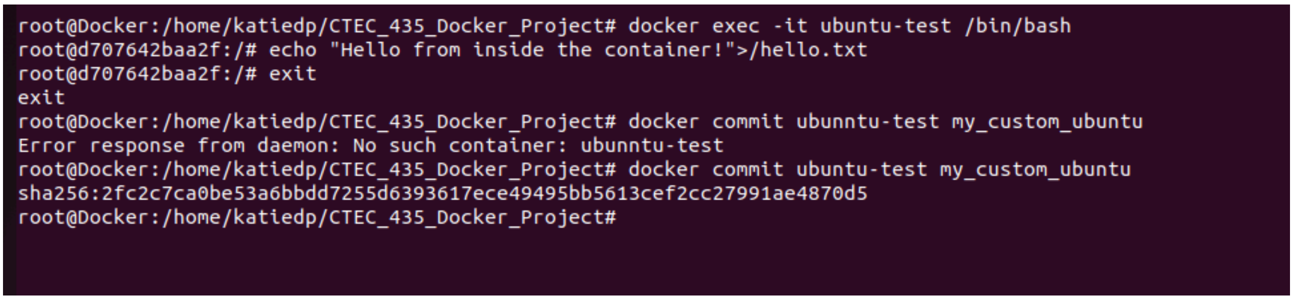
docker exec -it ubuntu-test /bin/bash



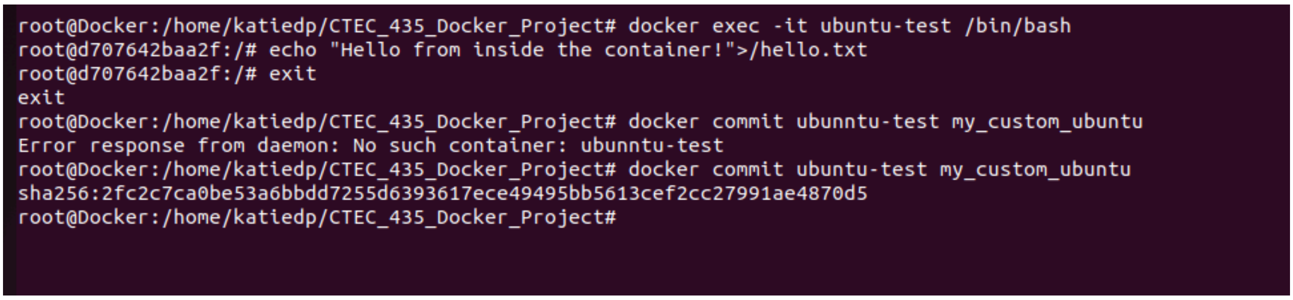
```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec -it ubuntu-test /bin/bash
root@d707642baa2f:/# echo "Hello from inside the container!">/hello.txt
root@d707642baa2f:/# exit
exit
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubunntu-test my_custom_ubuntu
Error response from daemon: No such container: ubunntu-test
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubuntu-test my_custom_ubuntu
sha256:2fc2c7ca0be53a6bbdd7255d6393617ece49495bb5613cef2cc27991ae4870d5
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

\`\`\`

- Create a new file inside the container:

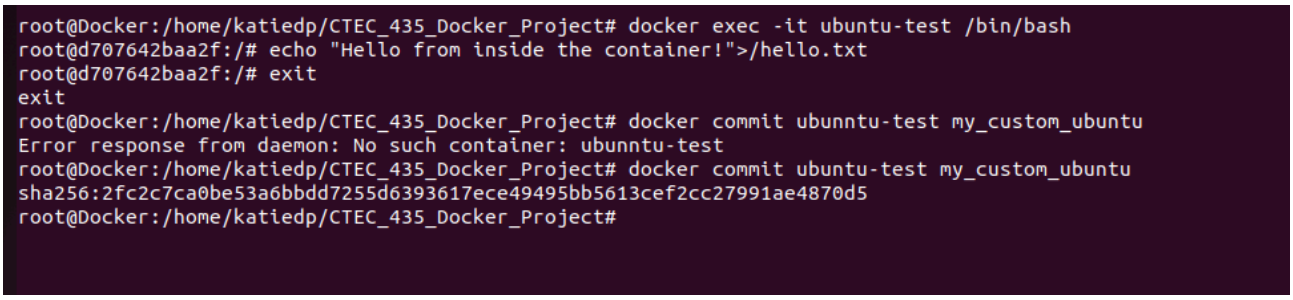


```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec -it ubuntu-test /bin/bash
root@d707642baa2f:/# echo "Hello from inside the container!">/hello.txt
root@d707642baa2f:/# exit
exit
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubunntu-test my_custom_ubuntu
Error response from daemon: No such container: ubunntu-test
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubuntu-test my_custom_ubuntu
sha256:2fc2c7ca0be53a6bbdd7255d6393617ece49495bb5613cef2cc27991ae4870d5
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

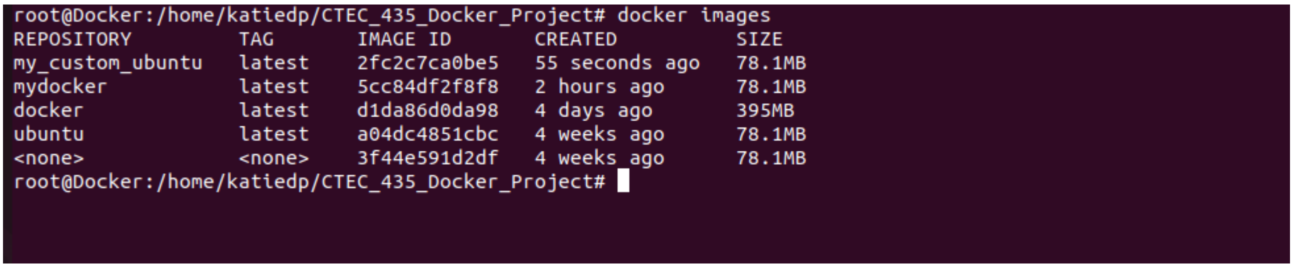
```

echo "Hello from inside the container!" > /hello.txt




```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec -it ubuntu-test /bin/bash
root@d707642baa2f:/# echo "Hello from inside the container!">/hello.txt
root@d707642baa2f:/# exit
exit
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubunntu-test my_custom_ubuntu
Error response from daemon: No such container: ubunntu-test
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubuntu-test my_custom_ubuntu
sha256:2fc2c7ca0be53a6bbdd7255d6393617ece49495bb5613cef2cc27991ae4870d5
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```


```

- Exit the container.

2. Commit Changes to a New Image:
   - Commit the changes made inside the container to a new Docker image:

```

docker commit ubuntu-test my_custom_ubuntu




```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec -it ubuntu-test /bin/bash
root@d707642baa2f:/# echo "Hello from inside the container!">/hello.txt
root@d707642baa2f:/# exit
exit
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubunntu-test my_custom_ubuntu
Error response from daemon: No such container: ubunntu-test
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubuntu-test my_custom_ubuntu
sha256:2fc2c7ca0be53a6bbdd7255d6393617ece49495bb5613cef2cc27991ae4870d5
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```


```

3. Verify the New Image:
   - List images using `docker images` to ensure the new image was created.



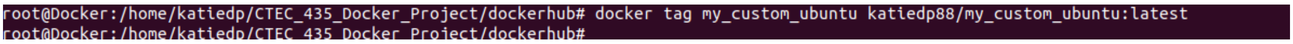
```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker images
REPOSITORY         TAG       IMAGE ID       CREATED          SIZE
my_custom_ubuntu   latest    2fc2c7ca0be5   55 seconds ago   78.1MB
mydocker           latest    5cc84df2f8f8   2 hours ago      78.1MB
docker             latest    d1da86d0da98   4 days ago       395MB
ubuntu             latest    a04dc4851cbc   4 weeks ago      78.1MB
<none>             <none>    3f44e591d2df   4 weeks ago      78.1MB
root@Docker:/home/katiedp/CTEC_435_Docker_Project# 
```

Task 8: Tagging and Pushing Custom Images to Docker Hub

1. Tag the Custom Image:

- Tag the newly created image for Docker Hub:

```
docker tag my_custom_ubuntu <dockerhub_username>/my_custom_ubuntu:latest
```

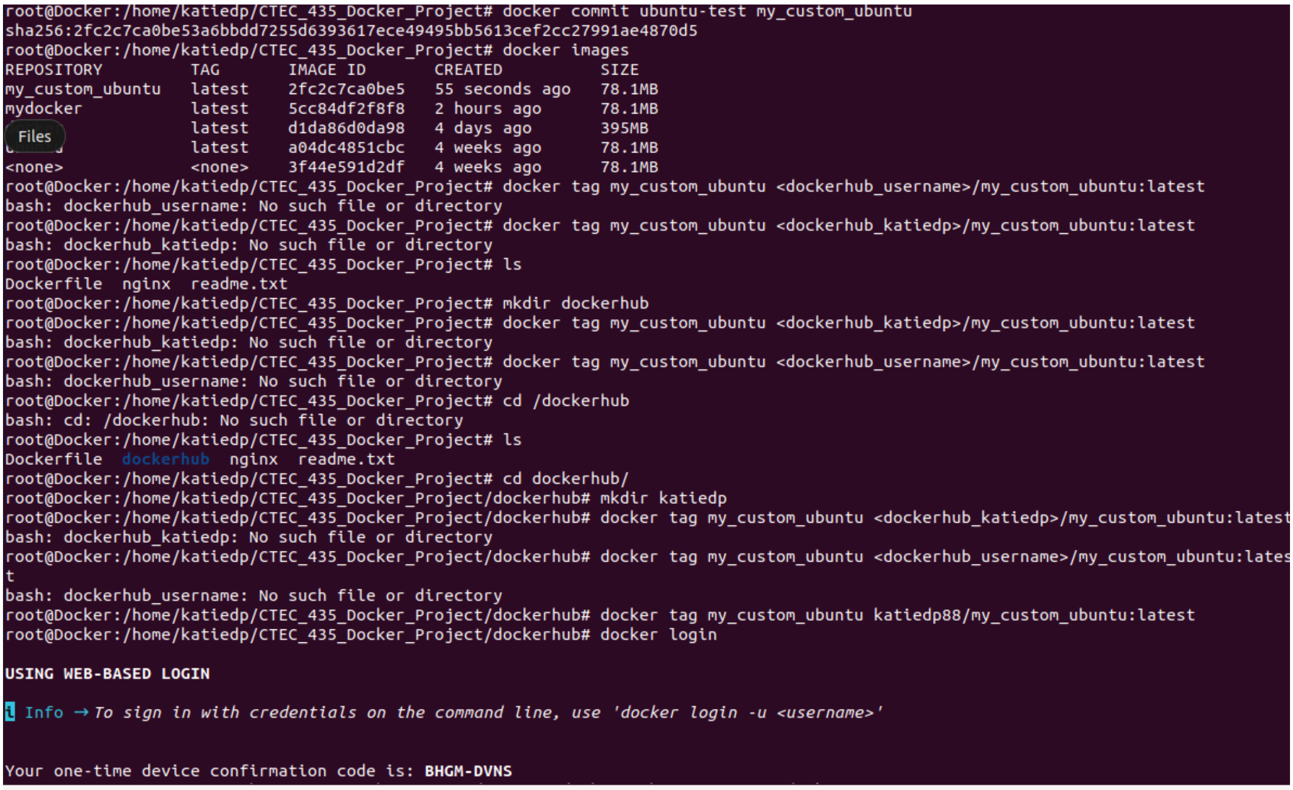


```
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker tag my_custom_ubuntu katiedp88/my_custom_ubuntu:latest
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub#
```

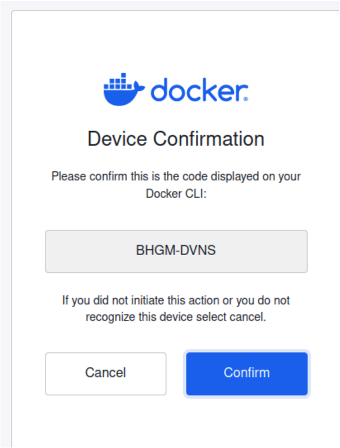
2. Push the Image to Docker Hub:

- Log in to Docker Hub and push the image:

```
docker login
docker push <dockerhub_username>/my_custom_ubuntu:latest
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker commit ubuntu-test my_custom_ubuntu
sha256:2fc2c7ca0be53a6bbdd7255d6393617ece49495bb5613cef2cc27991ae4870d5
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker images
REPOSITORY         TAG       IMAGE ID       CREATED          SIZE
my_custom_ubuntu   latest    2fc2c7ca0be5   55 seconds ago   78.1MB
mydocker           latest    5cc84df2f8f8   2 hours ago      78.1MB
                   latest    d1da86d0da98   4 days ago       395MB
 Files             latest    a04dc4851cbc   4 weeks ago      78.1MB
<none>             <none>    3f44e591d2df   4 weeks ago      78.1MB
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker tag my_custom_ubuntu <dockerhub_username>/my_custom_ubuntu:latest
bash: dockerhub_username: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker tag my_custom_ubuntu <dockerhub_katiedp>/my_custom_ubuntu:latest
bash: dockerhub_katiedp: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project# ls
Dockerfile  nginx  readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project# mkdir dockerhub
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker tag my_custom_ubuntu <dockerhub_katiedp>/my_custom_ubuntu:latest
bash: dockerhub_katiedp: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker tag my_custom_ubuntu <dockerhub_username>/my_custom_ubuntu:latest
bash: dockerhub_username: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project# cd /dockerhub
bash: cd: /dockerhub: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project# ls
Dockerfile  dockerhub  nginx  readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project# cd dockerhub/
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# mkdir katiedp
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker tag my_custom_ubuntu <dockerhub_katiedp>/my_custom_ubuntu:latest
bash: dockerhub_katiedp: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker tag my_custom_ubuntu <dockerhub_username>/my_custom_ubuntu:lates
t
bash: dockerhub_username: No such file or directory
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker tag my_custom_ubuntu katiedp88/my_custom_ubuntu:latest
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker login

USING WEB-BASED LOGIN

i Info → To sign in with credentials on the command line, use 'docker login -u <username>'


Your one-time device confirmation code is: BHGM-DVNS
```

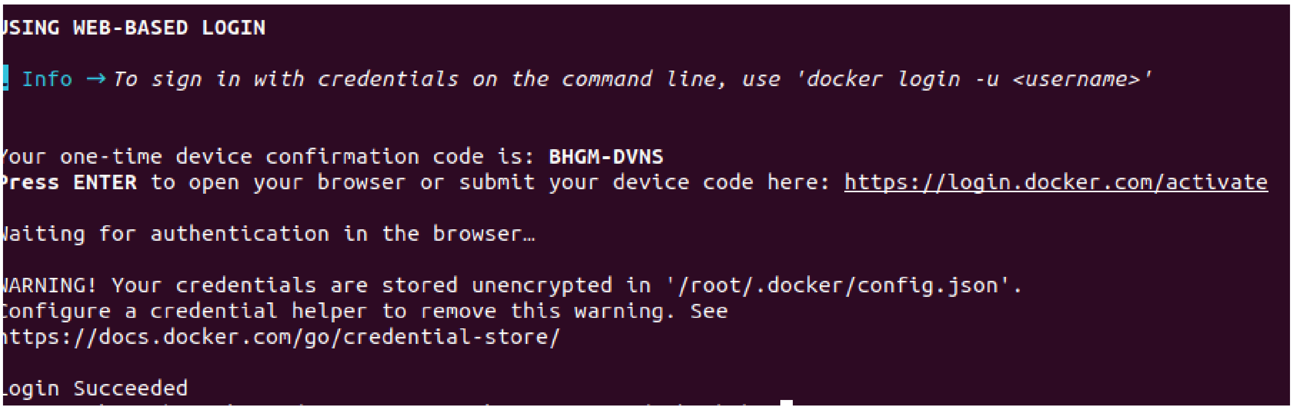


Device Confirmation

Please confirm this is the code displayed on your Docker CLI:

BHGM-DVNS

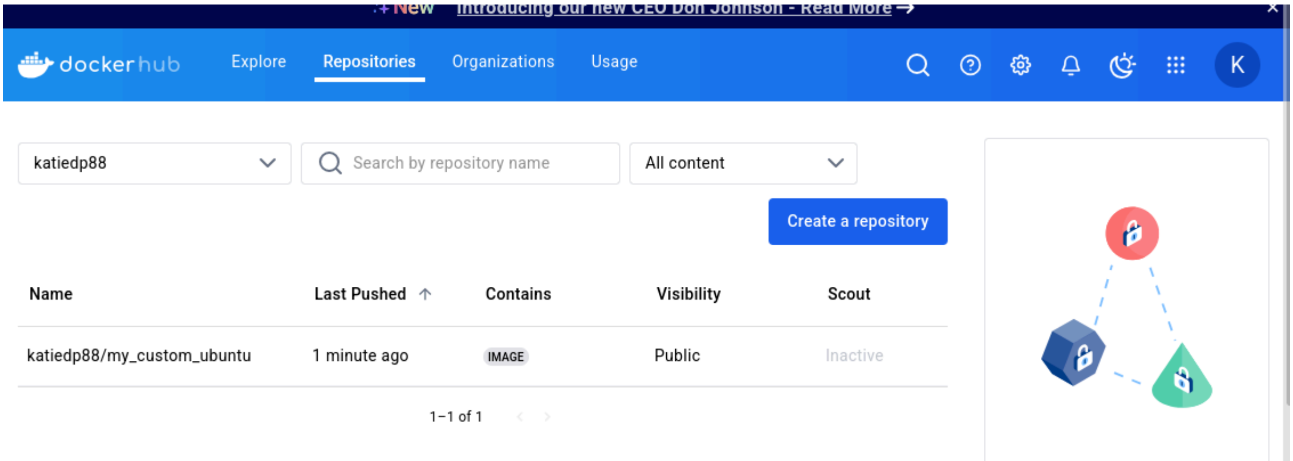
If you did not initiate this action or you do not recognize this device select cancel.



```
JSING WEB-BASED LOGIN

▍ Info → To sign in with credentials on the command line, use 'docker login -u <username>'

our one-time device confirmation code is: BHGM-DVNS
ress ENTER to open your browser or submit your device code here: https://login.docker.com/activate

aiting for authentication in the browser…

ARNING! Your credentials are stored unencrypted in '/root/.docker/config.json'.
onfigure a credential helper to remove this warning. See
ttps://docs.docker.com/go/credential-store/

ogin Succeeded
```

```
```

3. Verify the Image on Docker Hub:
  - Visit Docker Hub to confirm the image has been uploaded.



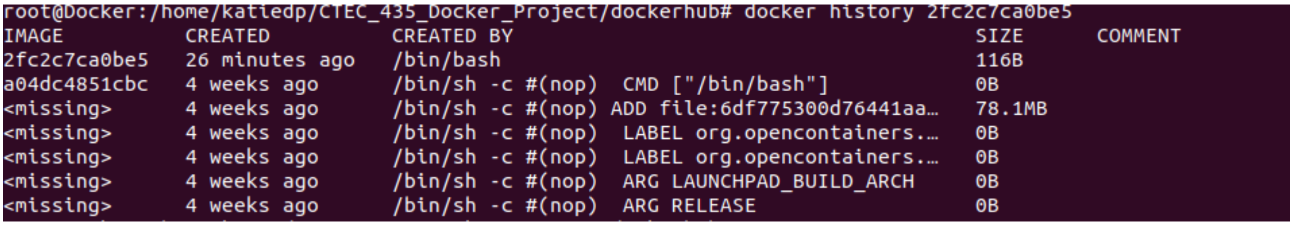
Name	Last Pushed ↑	Contains	Visibility	Scout
katiedp88/my_custom_ubuntu	1 minute ago	IMAGE	Public	Inactive

1–1 of 1

Task 9: Exploring Image Structure and Layers

1. Inspect Image Layers:
  - Use the `docker history` command to view the layers of your custom image:

```
docker history <image_id>
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project/dockerhub# docker history 2fc2c7ca0be5
IMAGE          CREATED          CREATED BY                                      SIZE      COMMENT
2fc2c7ca0be5   26 minutes ago   /bin/bash                                       116B
a04dc4851cbc   4 weeks ago      /bin/sh -c #(nop)  CMD ["/bin/bash"]            0B
<missing>      4 weeks ago      /bin/sh -c #(nop) ADD file:6df775300d76441aa…   78.1MB
<missing>      4 weeks ago      /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      4 weeks ago      /bin/sh -c #(nop)  LABEL org.opencontainers.…   0B
<missing>      4 weeks ago      /bin/sh -c #(nop)  ARG LAUNCHPAD_BUILD_ARCH     0B
<missing>      4 weeks ago      /bin/sh -c #(nop)  ARG RELEASE                  0B
```

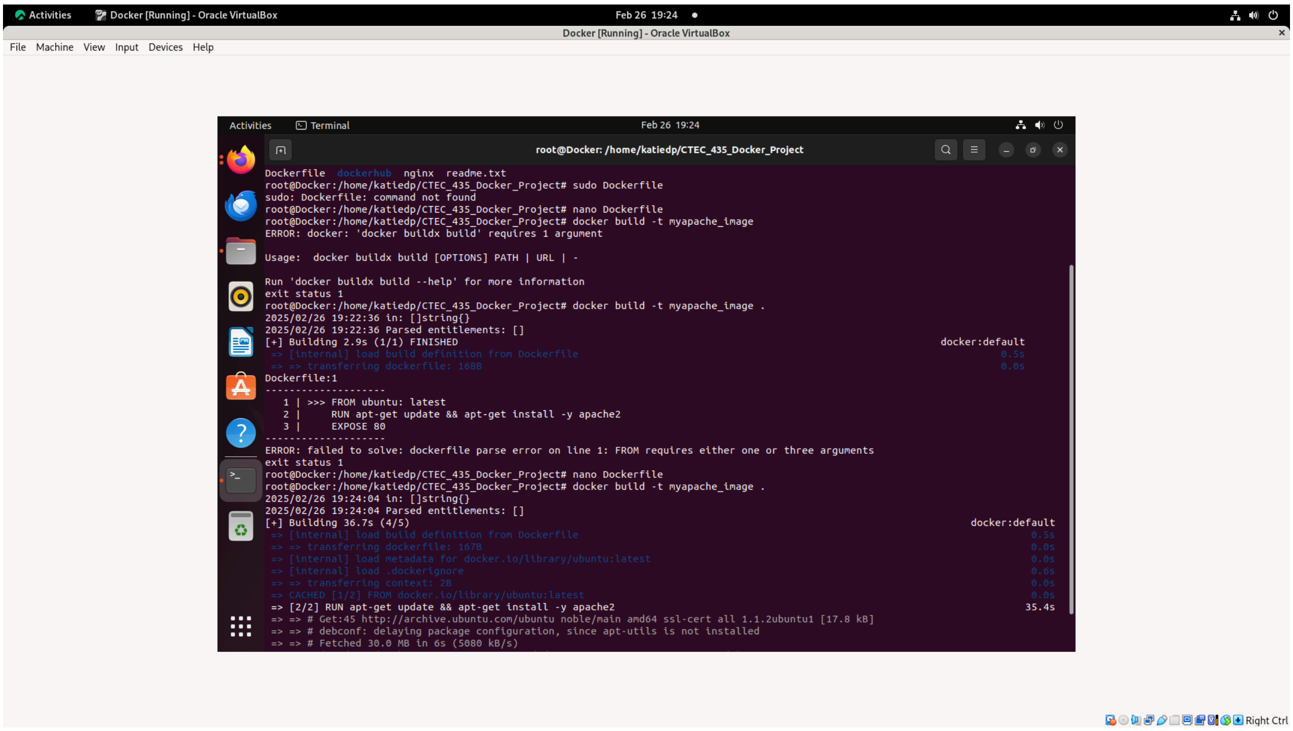
```
```

Part 5: Creating Custom Images Using Dockerfile

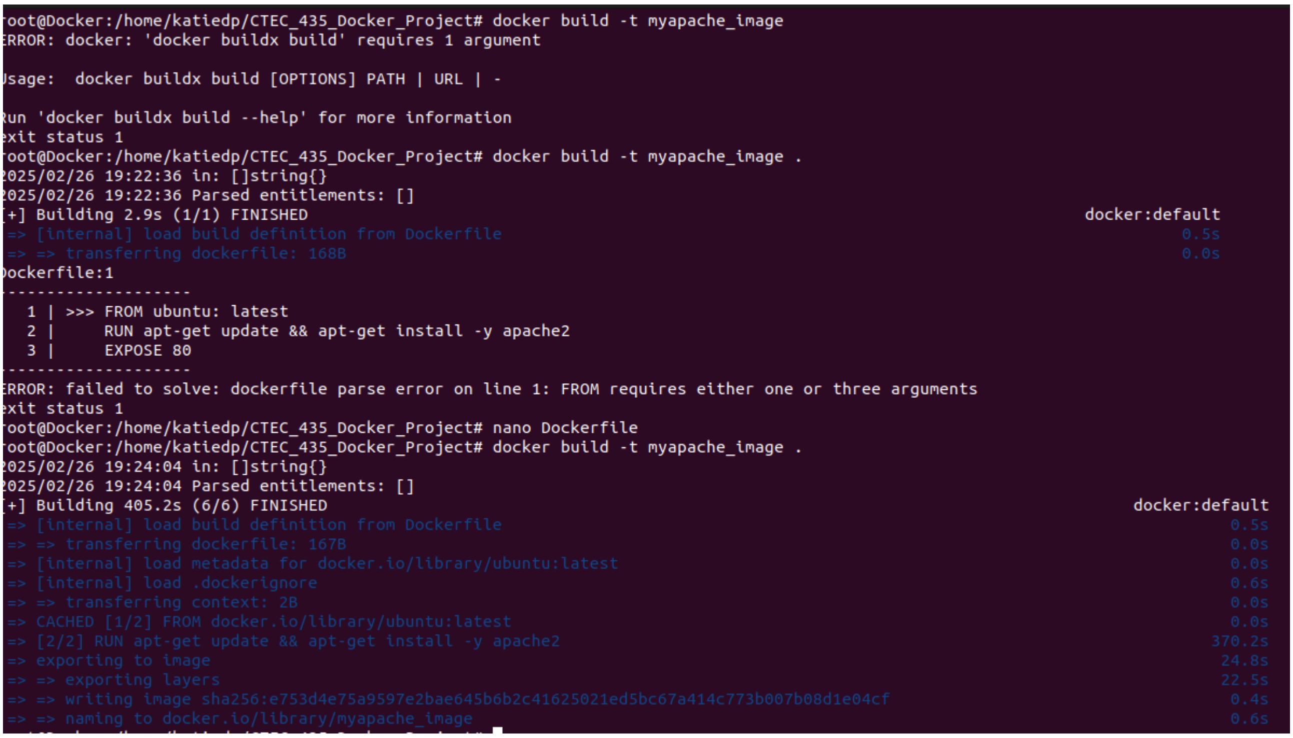
Task 10: Building a Custom Docker Image

1. Create a `Dockerfile`:
   - Write a `Dockerfile` that installs Apache on an Ubuntu base image:

```Dockerfile
FROM ubuntu:latest
RUN apt-get update && apt-get install -y apache2
EXPOSE 80
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```


Activities
Docker [Running] - Oracle VirtualBox
Feb 26 19:24
Docker [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Feb 26 19:24
root@Docker: /home/katiedp/CTEC_435_Docker_Project
Right Ctrl

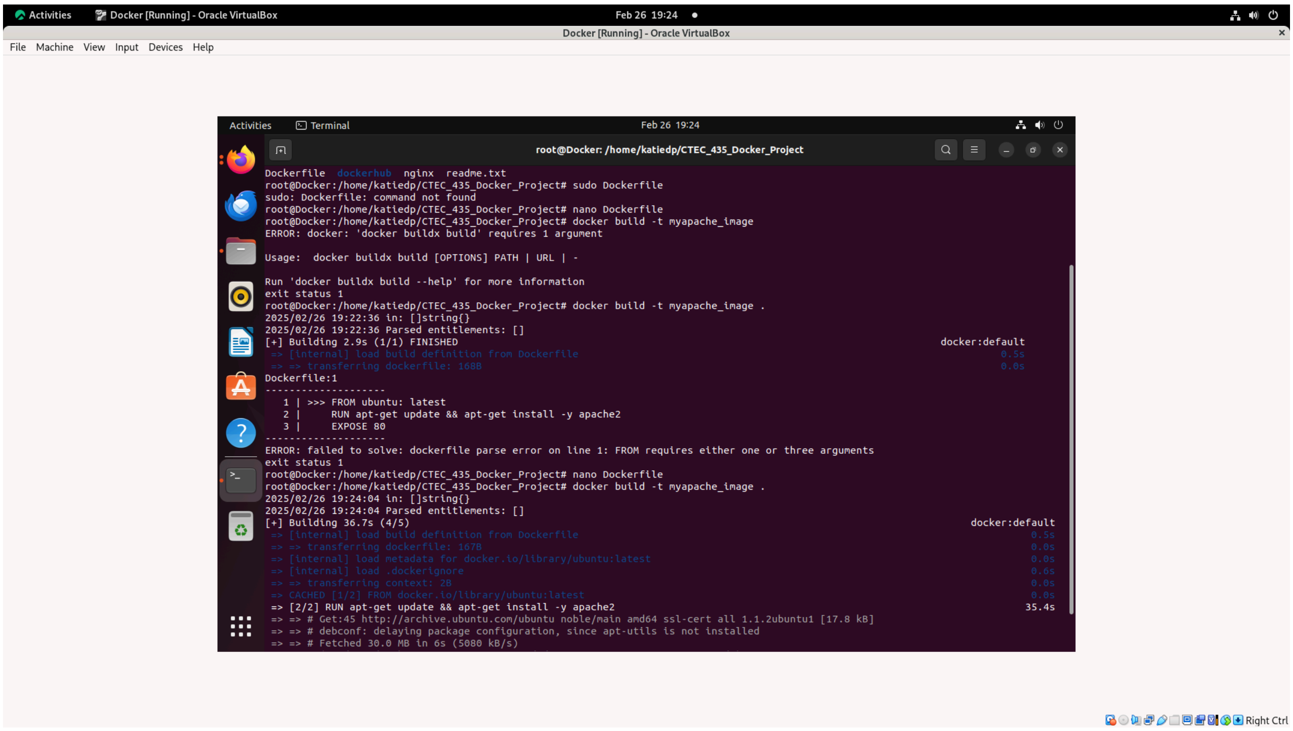
```
Dockerfile  dockerhub  nginx  readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project# sudo Dockerfile
sudo: Dockerfile: command not found
root@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfile
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image
ERROR: docker: 'docker buildx build' requires 1 argument

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Run 'docker buildx build --help' for more information
exit status 1
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:22:36 in: []string{}
2025/02/26 19:22:36 Parsed entitlements: []
[+] Building 2.9s (1/1) FINISHED                                                    docker:default
 => [internal] load build definition from Dockerfile                                          0.5s
 => => transferring dockerfile: 168B                                                          0.0s
Dockerfile:1
--------------------
   1 | >>> FROM ubuntu: latest
   2 |     RUN apt-get update && apt-get install -y apache2
   3 |     EXPOSE 80
--------------------
ERROR: failed to solve: dockerfile parse error on line 1: FROM requires either one or three arguments
exit status 1
root@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfile
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:24:04 in: []string{}
2025/02/26 19:24:04 Parsed entitlements: []
[+] Building 36.7s (4/5)                                                               docker:default
 => [internal] load build definition from Dockerfile                                             0.5s
 => => transferring dockerfile: 167B                                                             0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                 0.0s
 => [internal] load .dockerignore                                                                0.6s
 => => transferring context: 2B                                                                  0.0s
 => CACHED [1/2] FROM docker.io/library/ubuntu:latest                                            0.0s
 => [2/2] RUN apt-get update && apt-get install -y apache2                                      35.4s
 => => # Get:45 http://archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
 => => # debconf: delaying package configuration, since apt-utils is not installed
 => => # Fetched 30.0 MB in 6s (5080 kB/s)
```



```
oot@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image
RROR: docker: 'docker buildx build' requires 1 argument

Jsage:  docker buildx build [OPTIONS] PATH | URL | -

Run 'docker buildx build --help' for more information
exit status 1
oot@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:22:36 in: []string{}
2025/02/26 19:22:36 Parsed entitlements: []
[+] Building 2.9s (1/1) FINISHED                                                     docker:default
=> [internal] load build definition from Dockerfile                                            0.5s
=> => transferring dockerfile: 168B                                                             0.0s
Dockerfile:1
--------------------
   1 | >>> FROM ubuntu: latest
   2 |     RUN apt-get update && apt-get install -y apache2
   3 |     EXPOSE 80
--------------------
RROR: failed to solve: dockerfile parse error on line 1: FROM requires either one or three arguments
exit status 1
oot@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfile
oot@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:24:04 in: []string{}
2025/02/26 19:24:04 Parsed entitlements: []
[+] Building 405.2s (6/6) FINISHED                                                       docker:default
=> [internal] load build definition from Dockerfile                                                0.5s
=> => transferring dockerfile: 167B                                                                 0.0s
=> [internal] load metadata for docker.io/library/ubuntu:latest                                     0.0s
=> [internal] load .dockerignore                                                                    0.6s
=> => transferring context: 2B                                                                      0.0s
=> CACHED [1/2] FROM docker.io/library/ubuntu:latest                                                0.0s
=> [2/2] RUN apt-get update && apt-get install -y apache2                                         370.2s
=> exporting to image                                                                              24.8s
=> => exporting layers                                                                             22.5s
=> => writing image sha256:e753d4e75a9597e2bae645b6b2c41625021ed5bc67a414c773b007b08d1e04cf          0.4s
=> => naming to docker.io/library/myapache_image                                                    0.6s
```

```

2. Build the Image:

- Build the Docker image from the `Dockerfile`:

```

docker build -t my_apache_image .



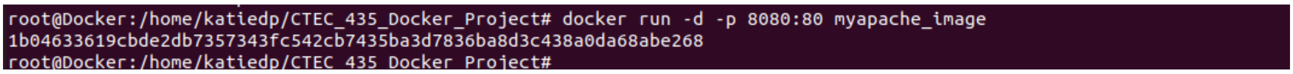
Activities
Docker [Running] - Oracle VirtualBox
Feb 26 19:24
Docker [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Feb 26 19:24
root@Docker: /home/katiedp/CTEC_435_Docker_Project
Right Ctrl

```
Dockerfile  dockerhub  nginx  readme.txt
root@Docker:/home/katiedp/CTEC_435_Docker_Project# sudo Dockerfile
sudo: Dockerfile: command not found
root@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfile
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image
ERROR: docker: 'docker buildx build' requires 1 argument

Usage:  docker buildx build [OPTIONS] PATH | URL | -

Run 'docker buildx build --help' for more information
exit status 1
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:22:36 in: []string{}
2025/02/26 19:22:36 Parsed entitlements: []
[+] Building 2.9s (1/1) FINISHED                                                 docker:default
 => [internal] load build definition from Dockerfile                                       0.5s
 => => transferring dockerfile: 168B                                                        0.0s
Dockerfile:1
--------------------
   1 | >>> FROM ubuntu: latest
   2 |     RUN apt-get update && apt-get install -y apache2
   3 |     EXPOSE 80
--------------------
ERROR: failed to solve: dockerfile parse error on line 1: FROM requires either one or three arguments
exit status 1
root@Docker:/home/katiedp/CTEC_435_Docker_Project# nano Dockerfile
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker build -t myapache_image .
2025/02/26 19:24:04 in: []string{}
2025/02/26 19:24:04 Parsed entitlements: []
[+] Building 36.7s (4/5)                                                             docker:default
 => [internal] load build definition from Dockerfile                                           0.5s
 => => transferring dockerfile: 167B                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                0.0s
 => [internal] load .dockerignore                                                               0.6s
 => => transferring context: 2B                                                                 0.0s
 => CACHED [1/2] FROM docker.io/library/ubuntu:latest                                           0.0s
 => [2/2] RUN apt-get update && apt-get install -y apache2                                     35.4s
 => => # Get:45 http://archive.ubuntu.com/ubuntu noble/main amd64 ssl-cert all 1.1.2ubuntu1 [17.8 kB]
 => => # debconf: delaying package configuration, since apt-utils is not installed
 => => # Fetched 30.0 MB in 6s (5080 kB/s)
```

```
```

3. Run the Custom Image:
   - Run the new Apache image:

```
docker run -d -p 8080:80 my_apache_image
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -d -p 8080:80 myapache_image
1b04633619cbde2db7357343fc542cb7435ba3d7836ba8d3c438a0da68abe268
root@Docker:/home/katiedp/CTEC_435_Docker_Project#
```

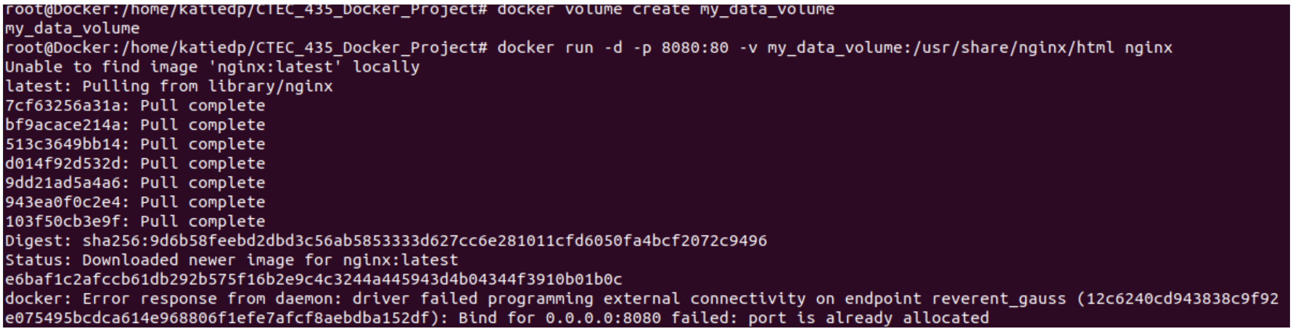
4. Access the Web Server:
   - Access the Apache web server by navigating to `http://<server_ip>:8080`.

Part 6: Persistent Data with Docker Volumes

Task 11: Creating and Managing Docker Volumes

1. Create a Docker Volume:
   - Create a new volume for persistent data storage:

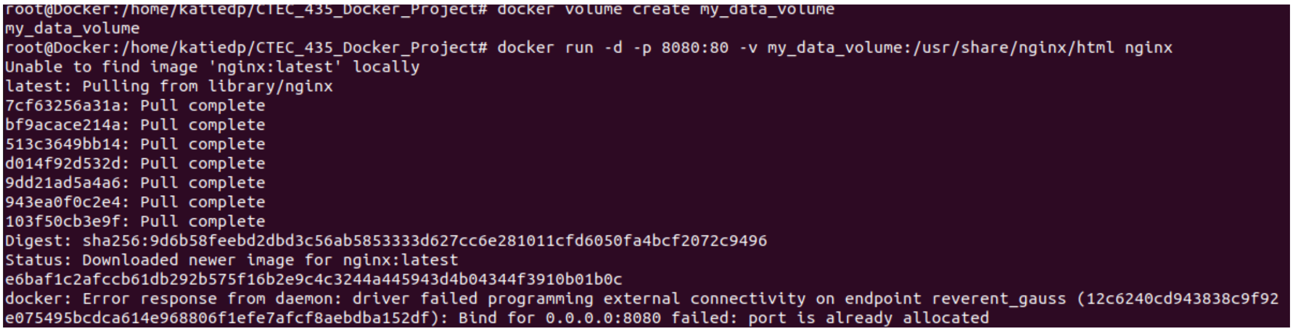
```
docker volume create my_data_volume
```



```
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker volume create my_data_volume
my_data_volume
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -d -p 8080:80 -v my_data_volume:/usr/share/nginx/html nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
e6baf1c2afccb61db292b575f16b2e9c4c3244a445943d4b04344f3910b01b0c
docker: Error response from daemon: driver failed programming external connectivity on endpoint reverent_gauss (12c6240cd943838c9f92
e075495bcdca614e968806f1efe7afcf8aebdba152df): Bind for 0.0.0.0:8080 failed: port is already allocated
```

2. Mount the Volume to a Container:
   - Run a container and mount the volume to persist data:

```
docker run -d -p 8081:80 -v my_data_volume:/usr/share/nginx/html nginx
```

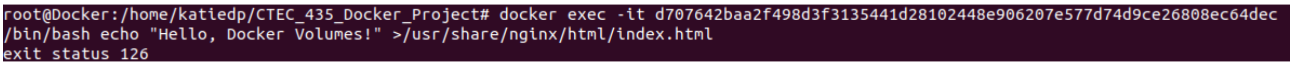


~~~
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker volume create my_data_volume
my_data_volume
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker run -d -p 8080:80 -v my_data_volume:/usr/share/nginx/html nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
e6baf1c2afccb61db292b575f16b2e9c4c3244a445943d4b04344f3910b01b0c
docker: Error response from daemon: driver failed programming external connectivity on endpoint reverent_gauss (12c6240cd943838c9f92
e075495bcdca614e968806f1efe7afcf8aebdba152df): Bind for 0.0.0.0:8080 failed: port is already allocated
~~~

```

3. Modify Data in the Volume:

- Enter the container and modify the contents of the mounted volume:

```

docker exec -it <container_id> /bin/bash

echo "Hello, Docker Volumes!" > /usr/share/nginx/html/index.html



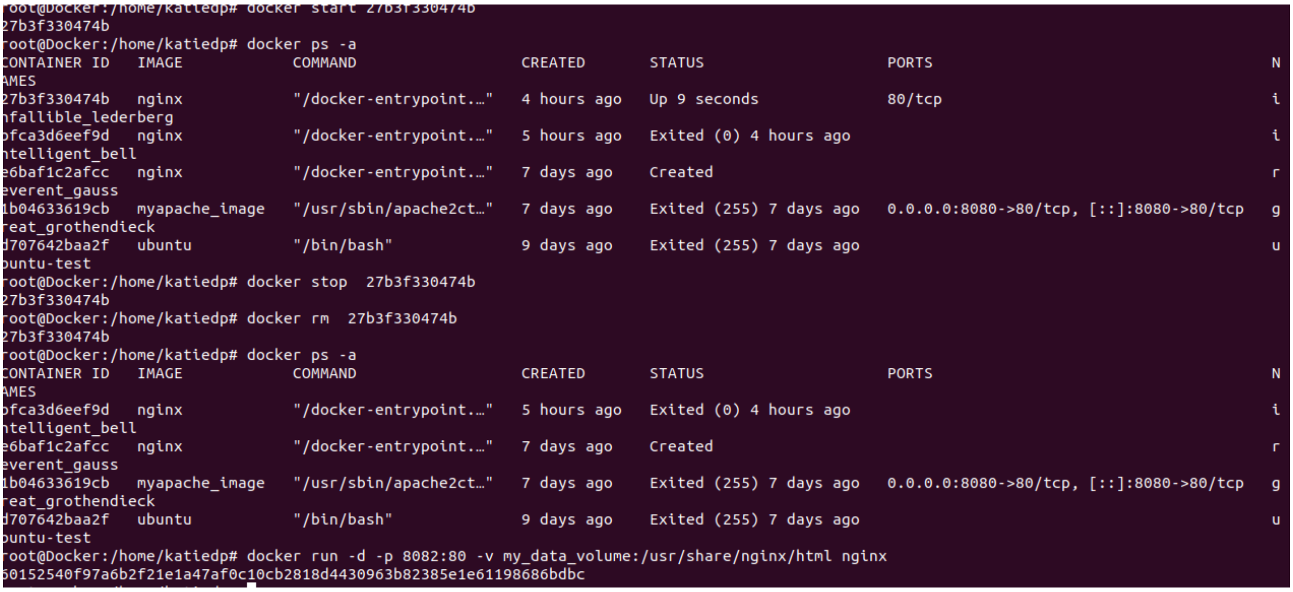
~~~
root@Docker:/home/katiedp/CTEC_435_Docker_Project# docker exec -it d707642baa2f498d3f3135441d28102448e906207e577d74d9ce26808ec64dec
/bin/bash echo "Hello, Docker Volumes!" >/usr/share/nginx/html/index.html
exit status 126
~~~

```

4. Verify Persistent Data:

- Stop and remove the container, then run a new container with the same volume and verify the data is still there:

```

docker stop <container_id>

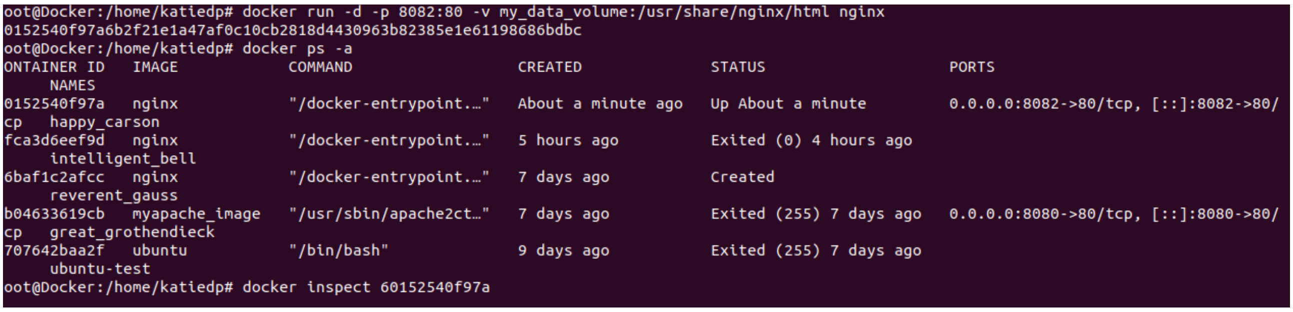
docker rm <container_id>

docker run -d -p 8082:80 -v my_data_volume:/usr/share/nginx/html nginx

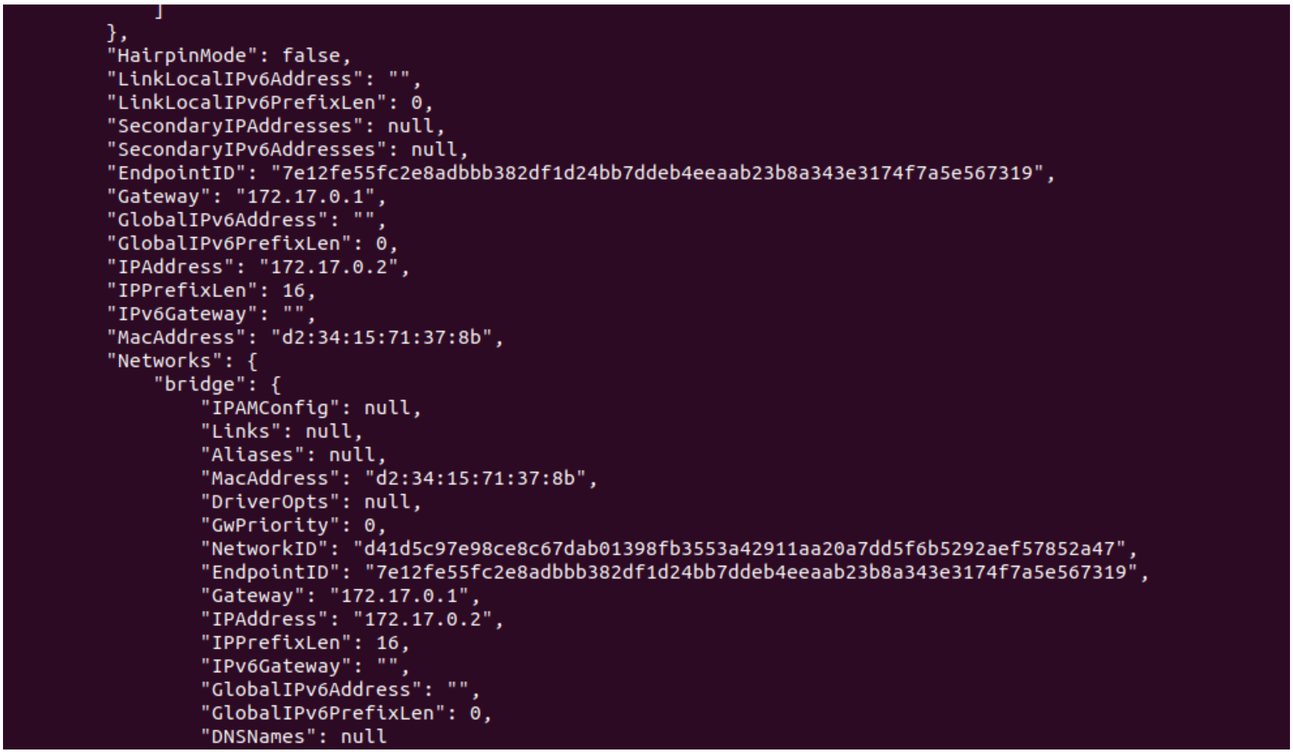


~~~
root@Docker:/home/katiedp# docker start 27b3f330474b
27b3f330474b
root@Docker:/home/katiedp# docker ps -a
CONTAINER ID   IMAGE            COMMAND                  CREATED       STATUS                    PORTS                                     N
AMES
27b3f330474b   nginx            "/docker-entrypoint.…"   4 hours ago   Up 9 seconds              80/tcp                                    i
nfallible_lederberg
bfca3d6eef9d   nginx            "/docker-entrypoint.…"   5 hours ago   Exited (0) 4 hours ago                                              i
ntelligent_bell
e6baf1c2afcc   nginx            "/docker-entrypoint.…"   7 days ago    Created                                                             r
everent_gauss
1b04633619cb   myapache_image   "/usr/sbin/apache2ct…"   7 days ago    Exited (255) 7 days ago   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   g
reat_grothendieck
d707642baa2f   ubuntu           "/bin/bash"              9 days ago    Exited (255) 7 days ago                                             u
buntu-test
root@Docker:/home/katiedp# docker stop  27b3f330474b
27b3f330474b
root@Docker:/home/katiedp# docker rm  27b3f330474b
27b3f330474b
root@Docker:/home/katiedp# docker ps -a
CONTAINER ID   IMAGE            COMMAND                  CREATED       STATUS                    PORTS                                     N
AMES
bfca3d6eef9d   nginx            "/docker-entrypoint.…"   5 hours ago   Exited (0) 4 hours ago                                              i
ntelligent_bell
e6baf1c2afcc   nginx            "/docker-entrypoint.…"   7 days ago    Created                                                             r
everent_gauss
1b04633619cb   myapache_image   "/usr/sbin/apache2ct…"   7 days ago    Exited (255) 7 days ago   0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   g
reat_grothendieck
d707642baa2f   ubuntu           "/bin/bash"              9 days ago    Exited (255) 7 days ago                                             u
buntu-test
root@Docker:/home/katiedp# docker run -d -p 8082:80 -v my_data_volume:/usr/share/nginx/html nginx
60152540f97a6b2f21e1a47af0c10cb2818d4430963b82385e1e61198686bdbc
~~~

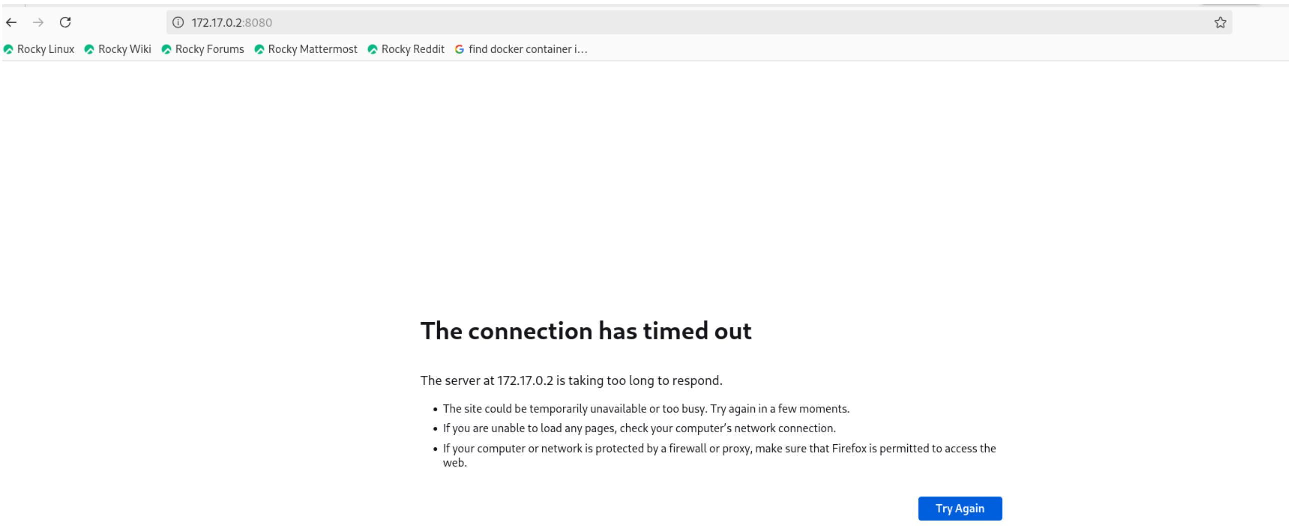
```

- Access `http://<server_ip>:8082` to verify that the data persists across container restarts.
```



```
oot@Docker:/home/katiedp# docker run -d -p 8082:80 -v my_data_volume:/usr/share/nginx/html nginx
0152540f97a6b2f21e1a47af0c10cb2818d4430963b82385e1e61198686bdbc
oot@Docker:/home/katiedp# docker ps -a
ONTAINER ID   IMAGE            COMMAND                  CREATED              STATUS                    PORTS
     NAMES
0152540f97a   nginx            "/docker-entrypoint.…"   About a minute ago   Up About a minute         0.0.0.0:8082->80/tcp, [::]:8082->80/
cp   happy_carson
fca3d6eef9d   nginx            "/docker-entrypoint.…"   5 hours ago          Exited (0) 4 hours ago
     intelligent_bell
6baf1c2afcc   nginx            "/docker-entrypoint.…"   7 days ago           Created
     reverent_gauss
b04633619cb   myapache_image   "/usr/sbin/apache2ct…"   7 days ago           Exited (255) 7 days ago   0.0.0.0:8080->80/tcp, [::]:8080->80/
cp   great_grothendieck
707642baa2f   ubuntu           "/bin/bash"              9 days ago           Exited (255) 7 days ago
     ubuntu-test
oot@Docker:/home/katiedp# docker inspect 60152540f97a
```



```
            ]
        },
        "HairpinMode": false,
        "LinkLocalIPv6Address": "",
        "LinkLocalIPv6PrefixLen": 0,
        "SecondaryIPAddresses": null,
        "SecondaryIPv6Addresses": null,
        "EndpointID": "7e12fe55fc2e8adbbb382df1d24bb7ddeb4eeaab23b8a343e3174f7a5e567319",
        "Gateway": "172.17.0.1",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "MacAddress": "d2:34:15:71:37:8b",
        "Networks": {
            "bridge": {
                "IPAMConfig": null,
                "Links": null,
                "Aliases": null,
                "MacAddress": "d2:34:15:71:37:8b",
                "DriverOpts": null,
                "GwPriority": 0,
                "NetworkID": "d41d5c97e98ce8c67dab01398fb3553a42911aa20a7dd5f6b5292aef57852a47",
                "EndpointID": "7e12fe55fc2e8adbbb382df1d24bb7ddeb4eeaab23b8a343e3174f7a5e567319",
                "Gateway": "172.17.0.1",
                "IPAddress": "172.17.0.2",
                "IPPrefixLen": 16,
                "IPv6Gateway": "",
                "GlobalIPv6Address": "",
                "GlobalIPv6PrefixLen": 0,
                "DNSNames": null
```



The connection has timed out

The server at 172.17.0.2 is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the web.