**Submitted to :Mr. Mahaz Khan**

# AIR UNIVERSITY, ISLAMABAD

## Department of Cyber Security

Secure Software Design & Development Lab (CY- 256L)

REG ID: 231312

# LAB TASK XII

Kiran Hashmi

BS-CYS 4 A

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Table of Contents


**Learning & Securing Docker** ..... 3

Docker platform: ..... 3

Containers ..... 3

Difference Between Container and Virtual Machine: ..... 3

**Docker Installation and Usage on Ubuntu** ..... 4

Step 1: ..... 4

Install Docker using the apt repository ..... 4

STEP 2: ..... 5

Playing with Busybox ..... 5

Docker Run ..... 6

**Webapps With Docker** ..... 7

Docker stopped ..... 9

**Docker Images** ..... 9

**My First Image** ..... 10

**Dockerfile** ..... 10

Creating a docker file: ..... 10

Making an account on docker Hub: ..... 11

**Docker push** ..... 12

Logging into docker ..... 12

Logged into Docker ..... 13

My docker hub repositories: ..... 14

**Securing Docker:** ..... 15

- Check that containers cant access host processes **(Rule 2)** ..... 15
- Check user namespace isolation **(Rule 3)** ..... 15
- Read-Only Filesystem **(Rule #8)** ..... 15
- Scan for Vulnerabilities **(Rule #9)** ..... 15

Installing Trivy ..... 15

Scanning ..... 16


**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Learning & Securing Docker

## Docker platform:

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security let you to run many containers simultaneously on a given host.

## Containers

Containers are lightweight and contain everything needed to run the application, so you dont need to rely on whats installed on the host. You can share containers while you work, and be sure that everyone you share with gets the same container that works in the same way.

## Difference Between Container and Virtual Machine:

| CONTAINERS | VIRTUAL MACHINES |
|---|---|
| ➢ Containers are an abstraction at the app layer that packages code and dependencies together.<br>➢ Multiple containers can run on the same machine and share the OS kernel with other containers.<br>➢ each running as isolated processes in user space<br>➢ Containers take up less space than<br>➢ VMs (container images are typically tens of MBs in size)<br>➢ can handle more applications and require fewer VMs and Operating systems. | ➢ Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.<br>➢ The hypervisor allows multiple VMs to run on a single machine.<br>➢ Each VM includes a<br>➢ full copy of an operating system, the application, necessary binaries and libraries<br>➢ taking up tens of GBs.<br>➢ VMs can also be slow to boot. |

**Date of Submission: 23/05/2025**

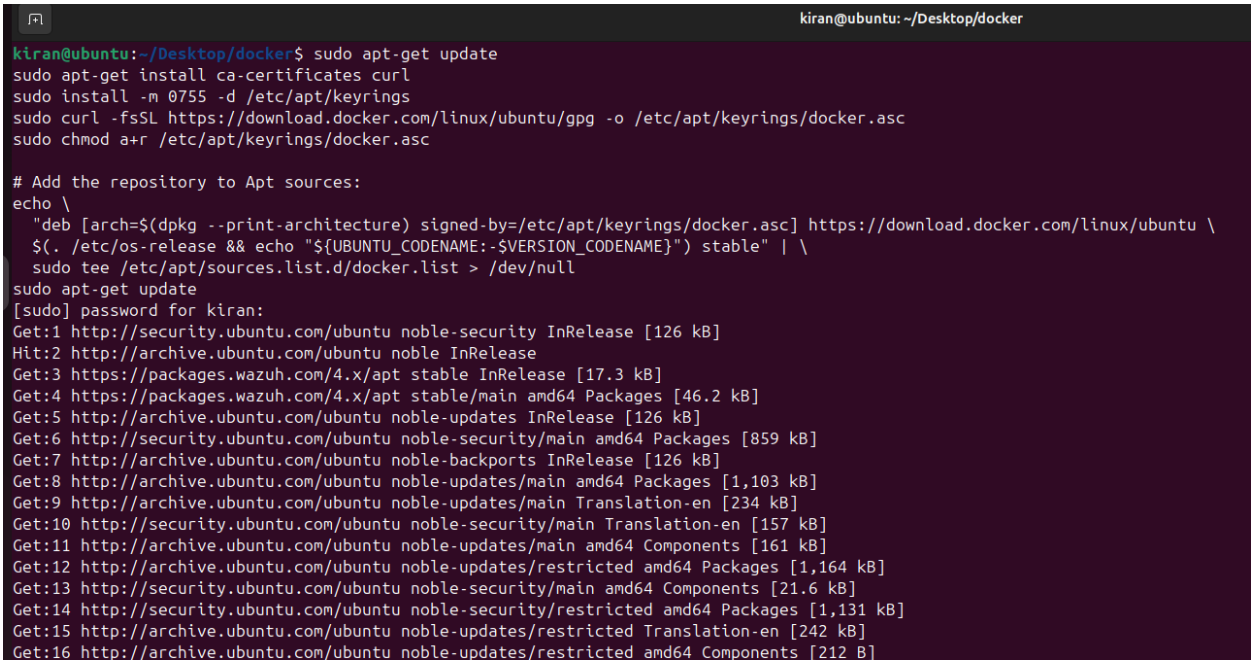
**Submitted to :Mr. Mahaz Khan**

# Docker Installation and Usage on Ubuntu

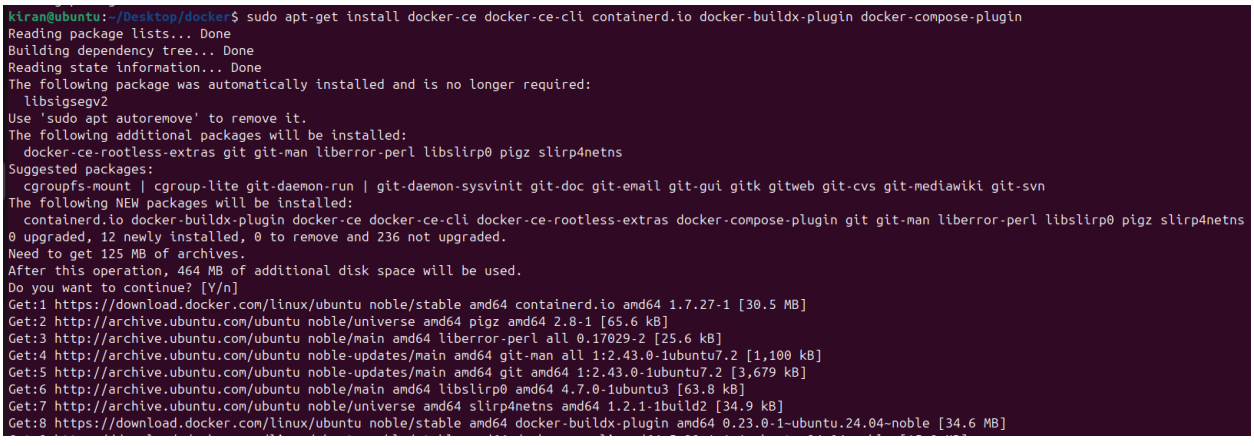
## Step 1:

### Install Docker using the apt repository

1. Set up Dockers apt repository.




2. Install the Docker packages. To install the latest version, run:




3. Verify that the Docker Engine installation is successful by running the hello-world image.

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

```
kiran@ubuntu:~/Desktop/docker$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:dd01f97f252193ae3210da231b1dca0cffab4aadb3566692d6730bf93f123a48
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

## STEP 2:

### Playing with Busybox

```
kiran@ubuntu:~/Desktop/docker$ sudo su
root@ubuntu:/home/kiran/Desktop/docker# docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
90b9666d4aed: Pull complete
Digest: sha256:3308bdfbc80b8e960219232df14f233a3c56979f392f56b0d9a8bc290c7dfd76
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
root@ubuntu:/home/kiran/Desktop/docker#
```

The pull command fetches the busybox image from the Docker registry and saves it to our system.

**Date of Submission: 23/05/2025**

```
root@ubuntu:/home/kiran/Desktop/docker# docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    74cc54e27dc4   4 months ago    10.1kB
busybox       latest    6d3e4188a38a   7 months ago    4.28MB
root@ubuntu:/home/kiran/Desktop/docker#
```

You can use the docker images command to see a list of all images on your system

## Docker Run

```
root@ubuntu:/home/kiran/Desktop/docker# docker run busybox
root@ubuntu:/home/kiran/Desktop/docker# docker run busybox echo "hello 231312 from busybox"
hello 231312 from busybox
root@ubuntu:/home/kiran/Desktop/docker#
```

The docker ps command shows you all containers that are currently running.

```
root@ubuntu:/home/kiran/Desktop/docker# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ubuntu:/home/kiran/Desktop/docker#
```

Since no containers are running, we see a blank line.
Lets try a more useful variant: **docker ps -a**

```
root@ubuntu:/home/kiran/Desktop/docker# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ubuntu:/home/kiran/Desktop/docker# docker ps -a
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS                      PORTS     NAMES
9e98a4e3fd95   busybox       "echo 'hello 231312 …"   2 minutes ago    Exited (0) 2 minutes ago              epic_noyce
f5d0755f2b1c   busybox       "sh"                     2 minutes ago    Exited (0) 2 minutes ago              bold_bardeen
e3f9fdecbf27   busybox       "echo hello from bus…"   2 minutes ago    Exited (0) 2 minutes ago              modest_neumann
dbdb9efaa597   busybox       "echohello 231312 fr…"   5 minutes ago    Created                               agitated_curran
cd5ef7cfd105   busybox       "sh"                     6 minutes ago    Exited (0) 6 minutes ago              happy_kilby
55538cfff96f   hello-world   "/hello"                 12 minutes ago   Exited (0) 12 minutes ago             brave_noyce
root@ubuntu:/home/kiran/Desktop/docker#
```

Running the **run command** with the -it flags attaches us to an interactive tty in the container. Now we can run as many commands in the container as we want.

```
root@ubuntu:/home/kiran/Desktop/docker# docker run -it busybox sh
/ # ls
bin    dev    etc    home   lib    lib64  proc   root   sys    tmp    usr    var
/ # uptime
 04:32:56 up 28 min,  0 users,  load average: 0.77, 1.46, 1.42
/ # ^C

/ # exit
```

**Date of Submission: 23/05/2025**

as a rule of thumb, I **clean up containers** once im done with them. To do that, you can run the docker rm command. Just copy the container IDs from above and paste them alongside the command.

```
root@ubuntu:/home/kiran/Desktop/docker# docker rm f5d0755f2b1c
f5d0755f2b1c
```

On **deletion**, you should see the IDs echoed back to you. If you have a bunch of containers to delete in one go, copy-pasting IDs can be tedious. In that case, you can simply run

```
root@ubuntu:/home/kiran/Desktop/docker# docker rm $(docker ps -a -q -f status=exited)
1e4863b97227
9e98a4e3fd95
e3f9fdecbf27
cd5ef7cfd105
55538cfff96f
root@ubuntu:/home/kiran/Desktop/docker#
```

In later versions of Docker, the docker container **prune command** can be used to achieve the same effect.

```
root@ubuntu:/home/kiran/Desktop/docker# docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
dbdb9efaa597411f245ad1286db8840304e87b912581d735a6ea4fc4b4cd2134

Total reclaimed space: 0B
root@ubuntu:/home/kiran/Desktop/docker#
```

## Webapps With Docker

Lets begin. The image that we are going to use is a single-page website that I've already created for the purpose of this demo and hosted on the registry - prakhar1989/static-site. We can download and run the image directly in one go using docker run. As noted above, the --rm flag automatically removes the container when it exits and the -it flag specifies an interactive terminal which makes it easier to kill the container with Ctrl+C (on windows).

**Submitted to :Mr. Mahaz Khan**

```
root@ubuntu:/home/kiran/Desktop/docker# docker run --rm -it prakhar1989/static-site
Unable to find image 'prakhar1989/static-site:latest' locally
latest: Pulling from prakhar1989/static-site
d4bce7fd68df: Pull complete
a3ed95caeb02: Pull complete
573113c4751a: Pull complete
31917632be33: Pull complete
77e66f18af1c: Pull complete
df3f108f3ade: Pull complete
d7a279eb19f5: Pull complete
e798589c05c5: Pull complete
78eeaf458ae0: Pull complete
Digest: sha256:bb6907c8db9ac4c6cadb25162a979e286575cd8b27727c08c7fbaf30988534db
Status: Downloaded newer image for prakhar1989/static-site:latest
Nginx is running...
```

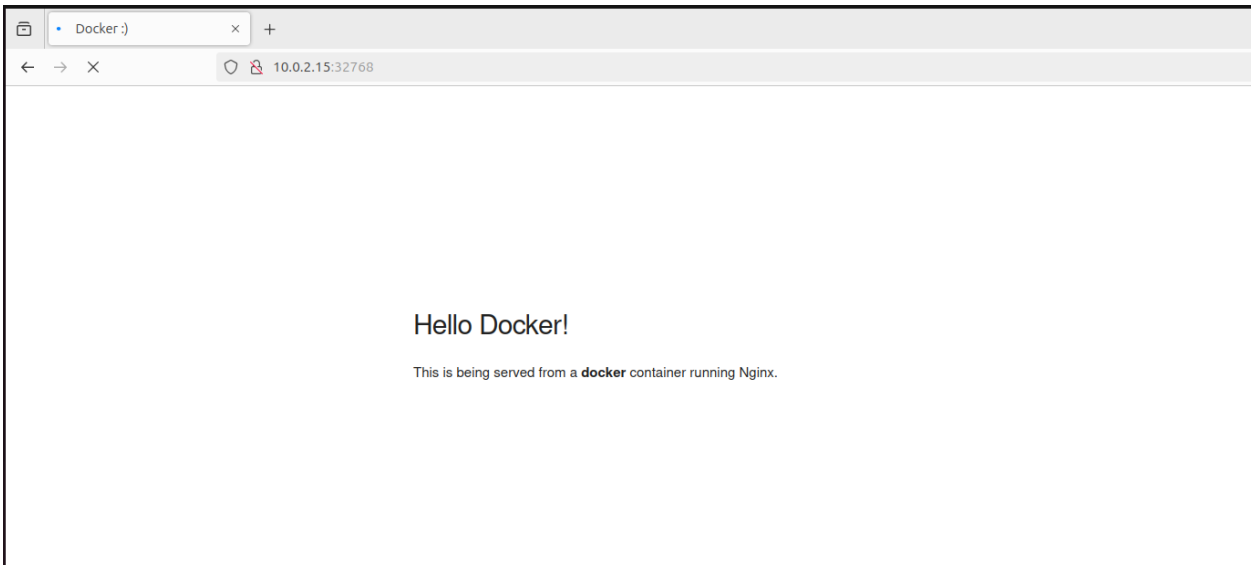
a Nginx is running... message

Now we can see the ports by running the docker port

```
root@ubuntu:/home/kiran/Desktop/docker# docker port static-site
80/tcp -> 0.0.0.0:32768
80/tcp -> [::]:32768
443/tcp -> 0.0.0.0:32769
443/tcp -> [::]:32769
root@ubuntu:/home/kiran/Desktop/docker#
```

**80/tcp 10.0.2.15:32768**

You can open **http://localhost:32768** in your browser.




To stop a detached container, run docker stop by giving the container ID. In this case, we can use the name static-site we used to start the container.

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

```
root@ubuntu:/home/kiran/Desktop/docker# docker stop static-site
static-site
root@ubuntu:/home/kiran/Desktop/docker#
```

**Docker stopped**


## Docker Images

```
root@ubuntu:/home/kiran/Desktop/docker# docker images
REPOSITORY                TAG       IMAGE ID       CREATED         SIZE
hello-world               latest    74cc54e27dc4   4 months ago    10.1kB
busybox                   latest    6d3e4188a38a   7 months ago    4.28MB
prakhar1989/static-site   latest    f01030e1dcf3   9 years ago     134MB
root@ubuntu:/home/kiran/Desktop/docker#
```

The above gives a list of images that Ive pulled from the registry, along with ones that Ive created myself (well shortly see how). The TAG refers to a particular snapshot of the image and the IMAGE ID is the corresponding unique identifier for that image.

you can pull a specific version of ubuntu image

**Date of Submission: 23/05/2025**

```
root@ubuntu:/home/kiran/Desktop/docker# docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
7c457f213c76: Pull complete
Digest: sha256:152dc042452c496007f07ca9127571cb9c29697f42acbfad72324b2bb2e43c98
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
root@ubuntu:/home/kiran/Desktop/docker#
```

## My First Image

goal in this section will be to create an image that sandboxes a simple Flask application. For the purposes of this workshop, there is a fun little Flask app that displays a random cat .gif

```
root@ubuntu:/home/kiran/Desktop/docker# git clone https://github.com/prakhar1989/docker-curriculum.git
Cloning into 'docker-curriculum'...
remote: Enumerating objects: 1737, done.
remote: Counting objects: 100% (72/72), done.
remote: Compressing objects: 100% (48/48), done.
Receiving objects: 100% (1737/1737), 9.16 MiB | 412.00 KiB/s, done.
remote: Total 1737 (delta 56), reused 24 (delta 24), pack-reused 1665 (from 4)
Resolving deltas: 100% (953/953), done.
root@ubuntu:/home/kiran/Desktop/docker# cd docker-curriculum/flask-app
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app#
```

## Dockerfile

A Dockerfile is a simple text file that contains a list of commands that the Docker client calls while creating an image. Its a simple way to automate the image creation process.

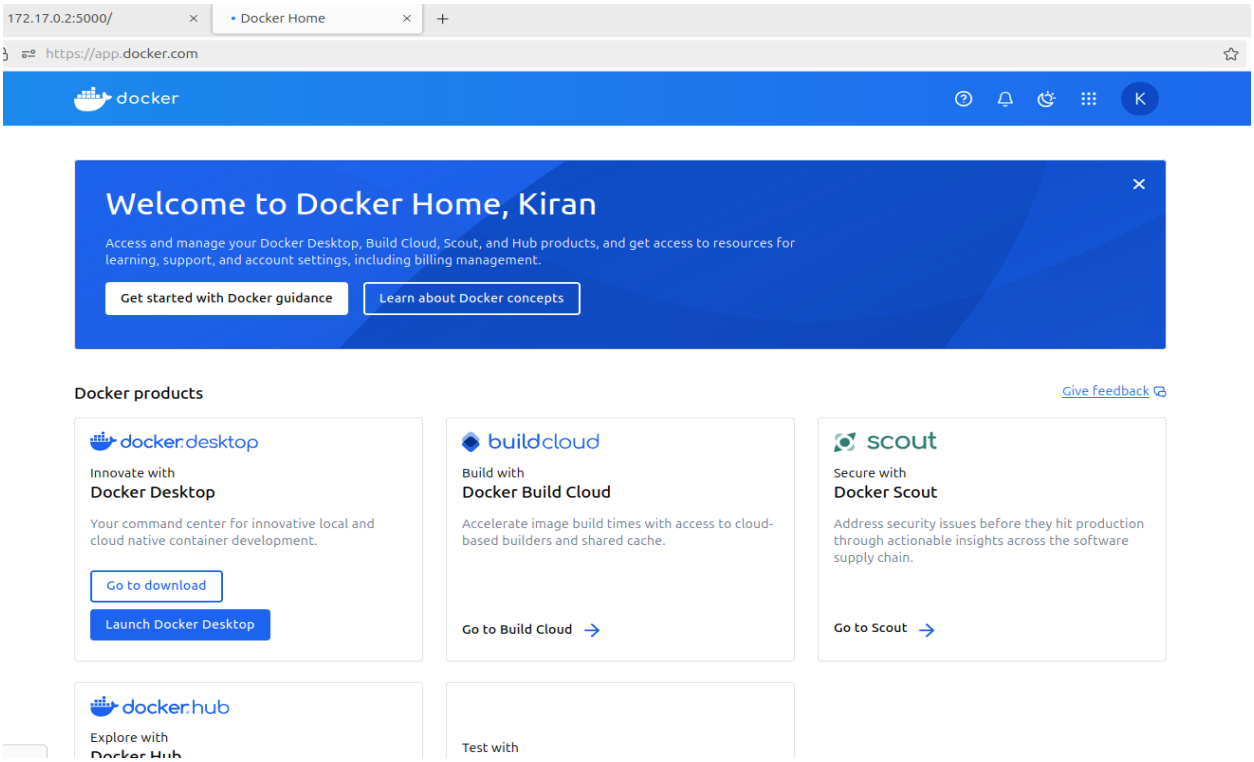
### Creating a docker file:

```
  GNU nano 7.2
# Use an official Python runtime as a parent image
FROM python:3.8

# Set the working directory in the container
WORKDIR /usr/src/app

# Copy the current directory contents into the container
COPY . .

# Install any needed packages specified in requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Make port 5000 available to the world outside this container
EXPOSE 5000

# Define environment variable (optional but good for Flask apps)
ENV FLASK_APP=app.py

# Run app.py when the container launches
CMD ["python", "app.py"]
```

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Making an account on docker Hub:




Now that we have our Dockerfile, we can build our image. The docker build command does the heavy-lifting of creating a Docker image from a Dockerfile.

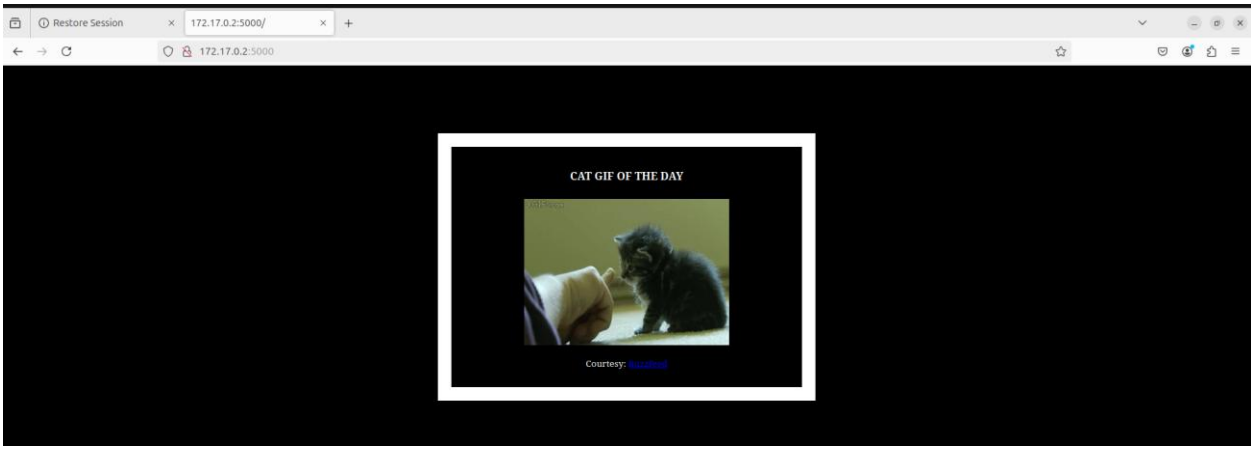
```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker build -t kiran231312/catnip .
[+] Building 9.4s (9/9) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 925B
 => [internal] load metadata for docker.io/library/python:3.8
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build context
 => => transferring context: 824B
 => [1/4] FROM docker.io/library/python:3.8@sha256:d411270700143fa2683cc8264d9fa5d3279fd3b6afff62ae81ea2f9d070e390c
 => CACHED [2/4] WORKDIR /usr/src/app
 => [3/4] COPY . .
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:556c488722a2e020822e02e7fbc085dfedbdaec5f633b0d82f9d856b6a392be1
 => => naming to docker.io/kiran231312/catnip
```

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum# echo "flask==2.0.3
werkzeug==2.0.3" > requirements.txt
```

Head over to the **URL with port 8888**, where your app should be live.

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**




# Docker push

lets use Docker Hub to publish the image.

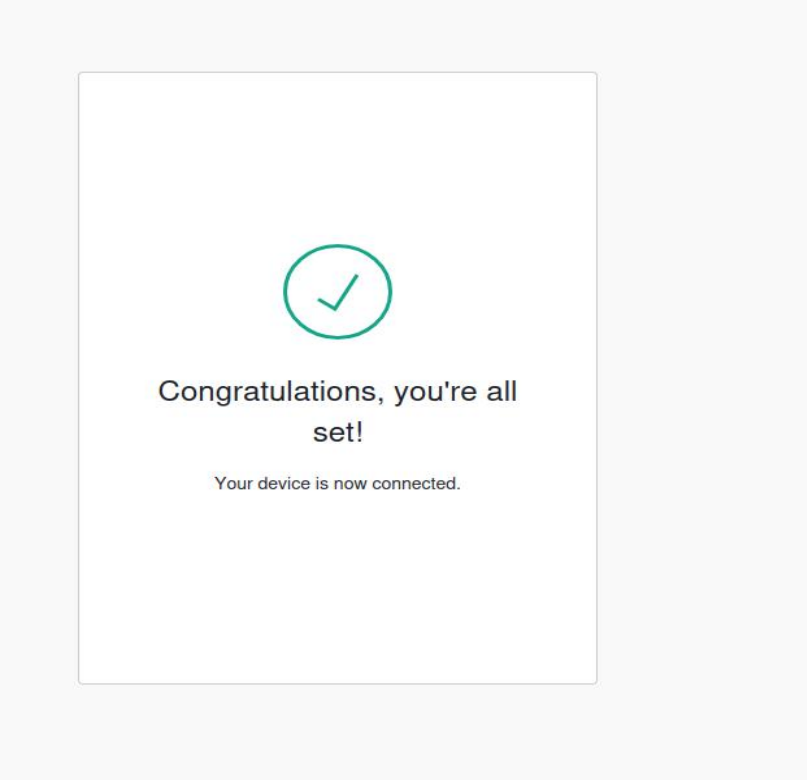
Logging into docker

```
kiran@ubuntu:~/Desktop/docker/docker-curriculum/flask-app$ 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: DVMF-LWWL
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate

Waiting for authentication in the browser…

WARNING! Your credentials are stored unencrypted in '/home/kiran/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
kiran@ubuntu:~/Desktop/docker/docker-curriculum/flask-app$
```

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Logged into Docker




- Move into flask directory
- Build image
- Push image

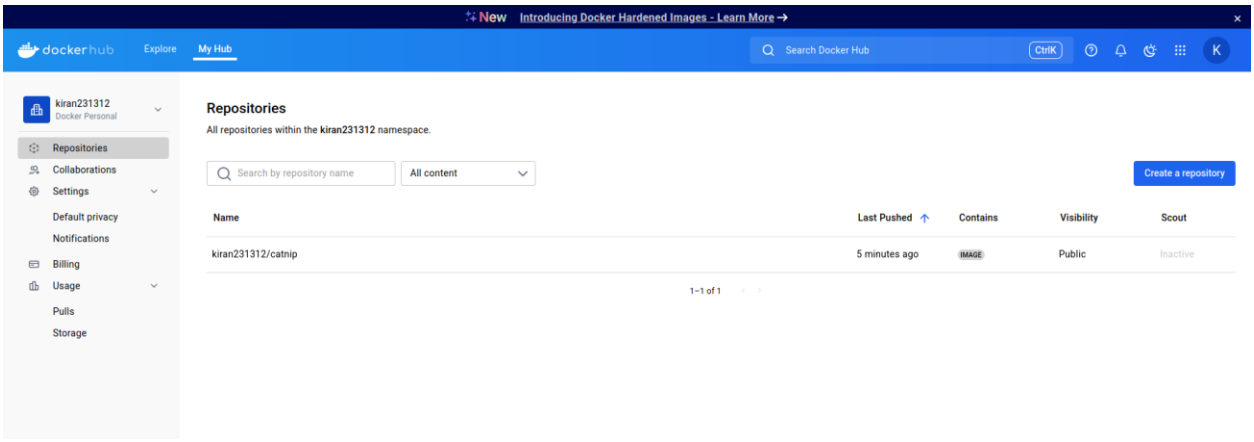
```
kiran@ubuntu:~/Desktop/docker$ cd docker-curriculum/flask-app
docker build -t kiran231312/catnip .
docker push kiran231312/catnip
[+] Building 8.4s (10/10) FINISHED                                  docker:default
 => [internal] load build definition from Dockerfile                          0.1s
 => => transferring dockerfile: 925B                                          0.0s
 => [internal] load metadata for docker.io/library/python:3.8                 2.3s
 => [auth] library/python:pull token for registry-1.docker.io                 0.0s
 => [internal] load .dockerignore                                             0.1s
 => => transferring context: 2B                                               0.0s
 => [1/4] FROM docker.io/library/python:3.8@sha256:d411270700143fa2683cc8     0.0s
 => [internal] load build context                                             0.1s
 => => transferring context: 5.08kB                                           0.0s
 => CACHED [2/4] WORKDIR /usr/src/app                                         0.0s
 => [3/4] COPY . .                                                            0.1s
 => [4/4] RUN pip install --no-cache-dir -r requirements.txt                  5.4s
 => exporting to image                                                        0.2s
 => => exporting layers                                                       0.2s
 => => writing image sha256:a4fc749be650bd02c2a4708c076a9934f17609b775587     0.0s
 => => naming to docker.io/kiran231312/catnip                                 0.0s
Using default tag: latest
The push refers to repository [docker.io/kiran231312/catnip]
011ebdd6a504: Pushed
6076e67a2dc3: Pushed
f4fc9d119b12: Pushed
32ee710ca3c7: Mounted from library/python
1767e4d52b5a: Mounted from library/python
45b98afd69b3: Mounted from library/python
2bce433c3a29: Mounted from library/python
f91dc7a486d9: Mounted from library/python
3e14a6961052: Mounted from library/python
d50132f2fe78: Mounted from library/python
latest: digest: sha256:ed5adb874b6a35da59961c12e50a540798b4377e02c0e1b069dbe35ed9a2d625 size: 2422
```

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

## My docker hub repositories:

- We can see that I have uploaded a catnip image




Once that is done, we can view our image on Docker Hub.


**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Securing Docker:

- Check that containers cant access host processes **(Rule 2)**

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker run --rm ubuntu ps aux
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
0622fac788ed: Pull complete
Digest: sha256:6015f66923d7afbc53558d7ccffd325d43b4e249f41a6e93eef074c9505d2233
Status: Downloaded newer image for ubuntu:latest
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1 16.6  0.1   7888  3912 ?        Rs   18:26   0:00 ps aux
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app#
```

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker run -u 1000:1000 ubuntu id
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu)
```

- Check user namespace isolation **(Rule 3)**

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker run --cap-drop ALL --cap-add NET_BIND_SER
VICE ubuntu
```

- Read-Only Filesystem **(Rule #8)**

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker run --read-only -v /tmp ubuntu touch /tmp/test
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# docker run --read-only ubuntu touch /test
touch: cannot touch '/test': Read-only file system
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app#
```

- Scan for Vulnerabilities **(Rule #9)**

## Installing Trivy

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# sudo apt-get install wget apt-transport-https gnupg lsb-release
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -
echo "deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/trivy.list
sudo apt-get update && sudo apt-get install trivy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
wget is already the newest version (1.21.4-1ubuntu4.1).
wget set to manually installed.
apt-transport-https is already the newest version (2.7.14build2).
gnupg is already the newest version (2.4.4-2ubuntu17.2).
lsb-release is already the newest version (12.0-2).
The following packages were automatically installed and are no longer required:
  libllvm17t64 libsigsegv2 python3-netifaces
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

**Date of Submission: 23/05/2025**

**Submitted to :Mr. Mahaz Khan**

# Scanning

- Found 39 vulnerabilities

```
root@ubuntu:/home/kiran/Desktop/docker/docker-curriculum/flask-app# trivy image ubuntu:22.04
2025-05-27T18:36:16Z    INFO    [vulndb] Need to update DB
2025-05-27T18:36:16Z    INFO    [vulndb] Downloading vulnerability DB...
2025-05-27T18:36:16Z    INFO    [vulndb] Downloading artifact...        repo="mirror.gcr.io/aquasec/trivy-db:2"
64.31 MiB / 64.31 MiB [-------------------------------------------------------------------------------------------------------] 100.00% 2.54 MiB p/s 26s
2025-05-27T18:36:44Z    INFO    [vulndb] Artifact successfully downloaded       repo="mirror.gcr.io/aquasec/trivy-db:2"
2025-05-27T18:36:44Z    INFO    [vuln] Vulnerability scanning is enabled
2025-05-27T18:36:44Z    INFO    [secret] Secret scanning is enabled
2025-05-27T18:36:44Z    INFO    [secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2025-05-27T18:36:44Z    INFO    [secret] Please see also https://trivy.dev/v0.62/docs/scanner/secret#recommendation for faster secret detection
2025-05-27T18:36:57Z    INFO    Detected OS     family="ubuntu" version="22.04"
2025-05-27T18:36:57Z    INFO    [ubuntu] Detecting vulnerabilities...   os_version="22.04" pkg_num=101
2025-05-27T18:36:57Z    INFO    Number of language-specific files       num=0

Report Summary

┌──────────────────────────────┬────────┬─────────────────┬─────────┐
│            Target            │  Type  │ Vulnerabilities │ Secrets │
├──────────────────────────────┼────────┼─────────────────┼─────────┤
│ ubuntu:22.04 (ubuntu 22.04)  │ ubuntu │       39        │    -    │
└──────────────────────────────┴────────┴─────────────────┴─────────┘

Legend:
- '-': Not scanned
- '0': Clean (no security findings detected)


ubuntu:22.04 (ubuntu 22.04)

Total: 39 (UNKNOWN: 0, LOW: 31, MEDIUM: 8, HIGH: 0, CRITICAL: 0)

┌──────────────┬────────────────┬──────────┬────────┬───────────────────┬───────────────┬──────────────────────────────────────────────────────────────┐
│   Library    │ Vulnerability  │ Severity │ Status │ Installed Version │ Fixed Version │                            Title                             │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ coreutils    │ CVE-2016-2781  │ LOW      │affected│ 8.32-4.1ubuntu1.2 │               │coreutils: Non-privileged session can escape to the parent    │
│              │                │          │        │                   │               │session in chroot                                             │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2016-2781                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ gcc-12-base  │ CVE-2022-27943 │          │        │12.3.0-1ubuntu1~22.04│             │binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows    │
│              │                │          │        │                   │               │stack exhaustion in demangle_const                            │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2022-27943                    │
│              ├────────────────┤          │        │                   ├───────────────┼──────────────────────────────────────────────────────────────┤
│              │ CVE-2023-4039  │          │        │                   │               │gcc: -fstack-protector fails to guard dynamic stack           │
│              │                │          │        │                   │               │allocations on ARM64                                          │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-4039                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ gpgv         │ CVE-2022-3219  │          │        │ 2.2.27-3ubuntu2.3 │               │gnupg: denial of service issue (resource consumption) using   │
│              │                │          │        │                   │               │compressed packets                                            │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libpcre2-8-0 │ CVE-2022-41409 │ LOW      │        │10.39-3ubuntu0.1   │               │pcre2: negative repeat value in a pcre2test subject line      │
│              │                │          │        │                   │               │leads to inifinite...                                         │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2022-41409                    │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libpcre3     │ CVE-2017-11164 │          │        │2:8.39-13ubuntu0.22.04.1│           │pcre: OP_KETRMAX feature in the match function in              │
│              │                │          │        │                   │               │pcre_exec.c                                                   │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2017-11164                    │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libssl3      │ CVE-2024-41996 │          │        │3.0.2-0ubuntu1.19  │               │openssl: remote attackers (from the client side) to trigger   │
│              │                │          │        │                   │               │unnecessarily expensive server-side...                        │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2024-41996                    │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libstdc++6   │ CVE-2022-27943 │          │        │12.3.0-1ubuntu1~22.04│             │binutils: libiberty/rust-demangle.c in GNU GCC 11.2 allows    │
│              │                │          │        │                   │               │stack exhaustion in demangle_const                            │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2022-27943                    │
│              ├────────────────┤          │        │                   ├───────────────┼──────────────────────────────────────────────────────────────┤
│              │ CVE-2023-4039  │          │        │                   │               │gcc: -fstack-protector fails to guard dynamic stack           │
│              │                │          │        │                   │               │allocations on ARM64                                          │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-4039                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libsystemd0  │ CVE-2023-7008  │          │        │249.11-0ubuntu3.15 │               │systemd-resolved: Unsigned name response in signed zone is    │
│              │                │          │        │                   │               │not refused when DNSSEC=yes...                                │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-7008                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libtinfo6    │ CVE-2023-45918 │          │        │6.3-2ubuntu0.1     │               │ncurses: NULL pointer dereference in tgetstr in               │
│              │                │          │        │                   │               │tinfo/lib_termcap.c                                           │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-45918                    │
│              ├────────────────┤          │        │                   ├───────────────┼──────────────────────────────────────────────────────────────┤
│              │ CVE-2023-50495 │          │        │                   │               │ncurses: segmentation fault via _nc_wrap_entry()              │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-50495                    │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libudev1     │ CVE-2023-7008  │          │        │249.11-0ubuntu3.15 │               │systemd-resolved: Unsigned name response in signed zone is    │
│              │                │          │        │                   │               │not refused when DNSSEC=yes...                                │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-7008                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ libzstd1     │ CVE-2022-4899  │          │        │1.4.8+dfsg-3build1 │               │zstd: mysql: buffer overrun in util.c                         │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2022-4899                     │
├──────────────┼────────────────┼──────────┼────────┼───────────────────┼───────────────┼──────────────────────────────────────────────────────────────┤
│ login        │ CVE-2023-29383 │          │        │1:4.8.1-2ubuntu2.2 │               │shadow: Improper input validation in shadow-utils package     │
│              │                │          │        │                   │               │utility chfn                                                  │
│              │                │          │        │                   │               │https://avd.aquasec.com/nvd/cve-2023-29383                    │
│              ├────────────────┤          │        │                   ├───────────────┼──────────────────────────────────────────────────────────────┤
│              │ CVE-2024-56433 │          │        │                   │               │shadow-utils: Default subordinate ID configuration in         │
│              │                │          │        │                   │               │/etc/login.defs could lead to compromise                      │
```

**Date of Submission: 23/05/2025**