Roll No: 67
NAME: HERAMB. R. PAWAR
SUB: SEPM
CLASS: D11AD

PRACTICAL 8

AIM: TO UNDERSTAND DOCKER ARCHITECTURE AND CONTAINER LIFECYCLE, INSTALL DOCKER AND EXECUTE DOCKER COMMANDS TO MANAGE IMAGE AND INTERACT WITH Contaners.

Problem Statement : The practical includes creating Docker Image, running Docker Containers, managing Containers lifecycle, etc

Therory:

DOCKER:
Docker is Linux based, open- Source Contanerization platform that Developer use to build, run and package applications for deployment using Containers. Unlike Virtual Machines, Docker Container offers;
* Os level abstraction with optimum resource Utilization
* Interoperability
* Efficient build and test
* Faster application execution
* Fundamentally, Docker Containers modularize.

an application's functionality into multiple components that allow deploying, testing or scaling them independently when needed

Take for instance a Docker containerized database of an application. With such a framework you can scale or maintain the database independently from other modules / components of the application without ~~[illegible]~~ impacting the workloads of other critical systems

# Components of Docker architecture:

Docker comprises the following different component within its core architecture:

A) Image
B) Containers
C) Registries
D) Docker Engine

A) Images:
Images are like blueprints containing instruction for creating a Docker container. Image define
→ Application dependencies
→ The process that should run when the application launches

You can get images from DockerHUB or create

Your own images by including specific instruction within a file called Docker File.

B) Containers:
Containers are live instances of images on which an application or its independent modules are run.
In an object oriented programming analogy an image is a class and container ~~is~~ is an instance of that class. This allows operational efficiency by allowing you to multiple container from a single image.

C) Registries:
A Docker registry is like a repository of Images.
The default registry is the Docker HUB, a public registry that stores public and offical images for different languages and platforms
By default, a request for an image from Docker is searched within the Docker HUB registry

You can also own a private registry and configu-re it to be the default source of images for your custom requirements.

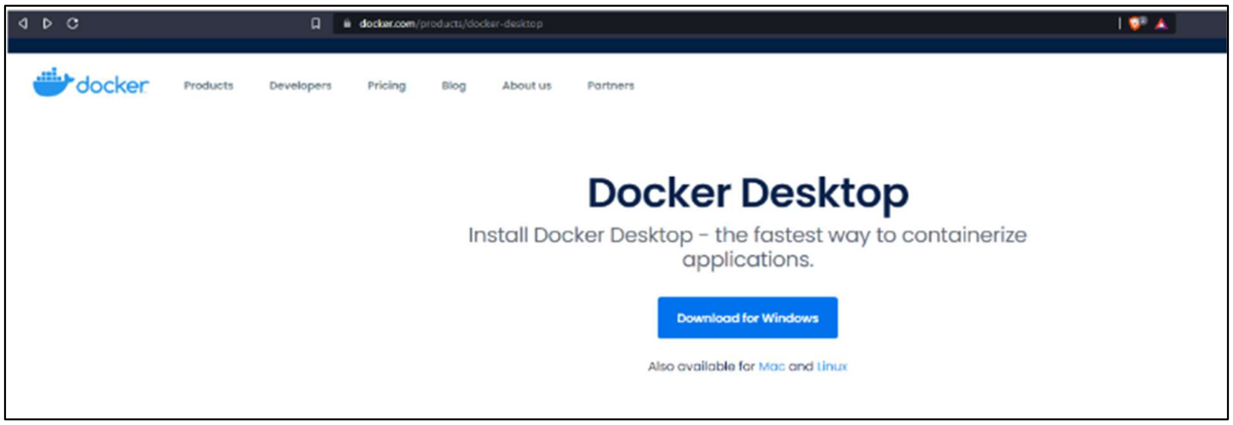
Conclusion:

Thus we learned how to install Docker on our machines and use basic Docker commands using the CLI, to create run and stop Docker containers

Name: Heramb Pawar **Subject**:SEPM **Roll:** 67

Practical Output:

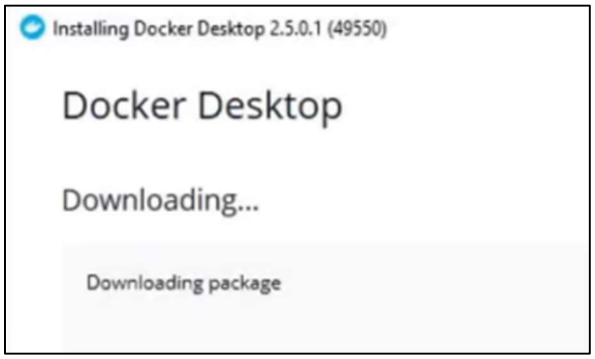
Downloading & Installing Docker:



Docker Desktop

Install Docker Desktop – the fastest way to containerize applications.

Also available for Mac and Linux

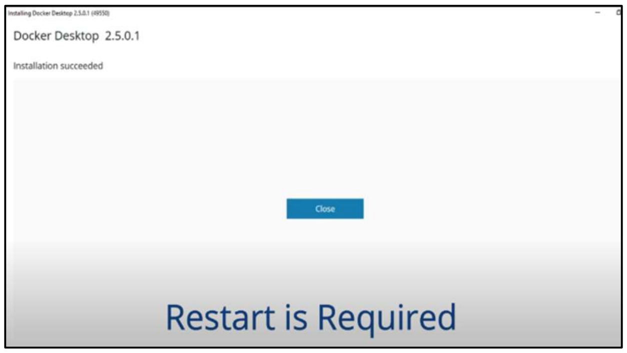


Installing Docker Desktop 2.5.0.1 (49550)

Docker Desktop

Downloading...

Downloading package

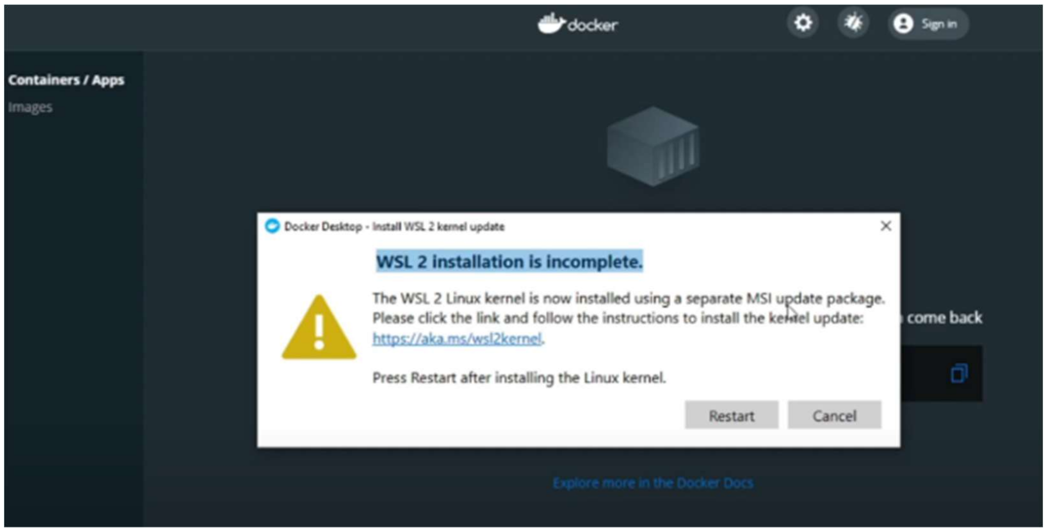


Docker Desktop 2.5.0.1

Installation succeeded

Restart is Required

Downloading And Installing WSL2



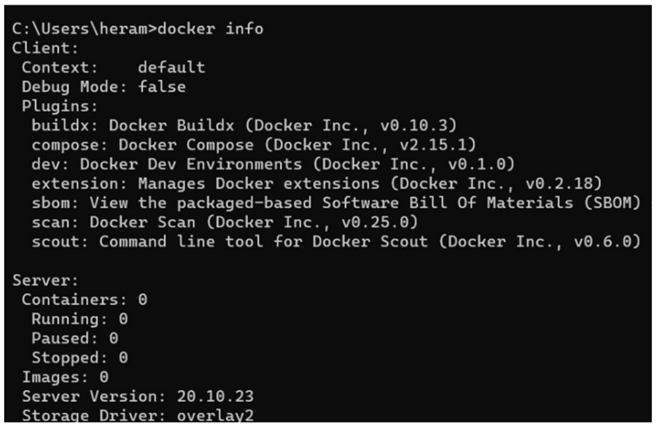
Docker Desktop - Install WSL 2 kernel update

WSL 2 installation is incomplete.

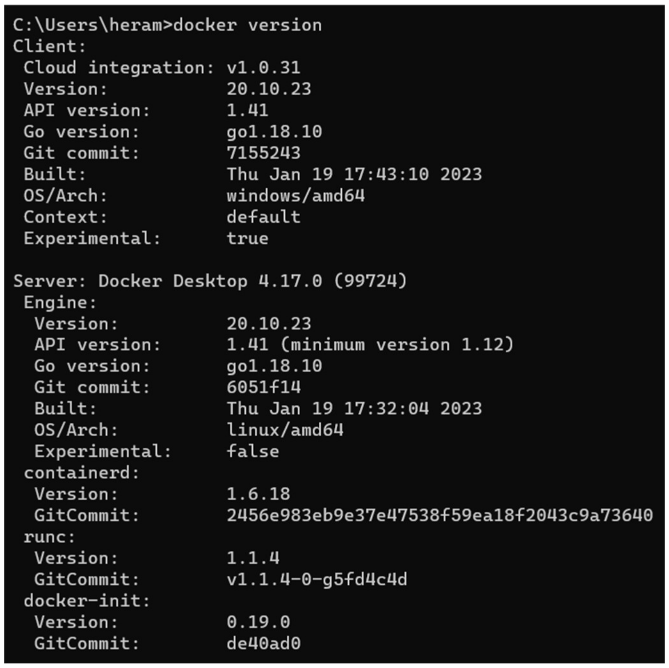
The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

Exploring Docker Commands:

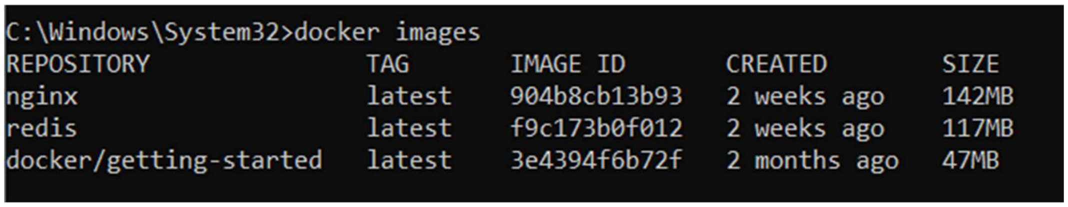


```
C:\Users\heram>docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc., v0.10.3)
  compose: Docker Compose (Docker Inc., v2.15.1)
  dev: Docker Dev Environments (Docker Inc., v0.1.0)
  extension: Manages Docker extensions (Docker Inc., v0.2.18)
  sbom: View the packaged-based Software Bill Of Materials (SBOM)
  scan: Docker Scan (Docker Inc., v0.25.0)
  scout: Command line tool for Docker Scout (Docker Inc., v0.6.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.23
 Storage Driver: overlay2
```



```
C:\Users\heram>docker version
Client:
 Cloud integration: v1.0.31
 Version:           20.10.23
 API version:       1.41
 Go version:        go1.18.10
 Git commit:        7155243
 Built:             Thu Jan 19 17:43:10 2023
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.17.0 (99724)
 Engine:
  Version:          20.10.23
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.10
  Git commit:       6051f14
  Built:            Thu Jan 19 17:32:04 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.18
  GitCommit:        2456e983eb9e37e47538f59ea18f2043c9a73640
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

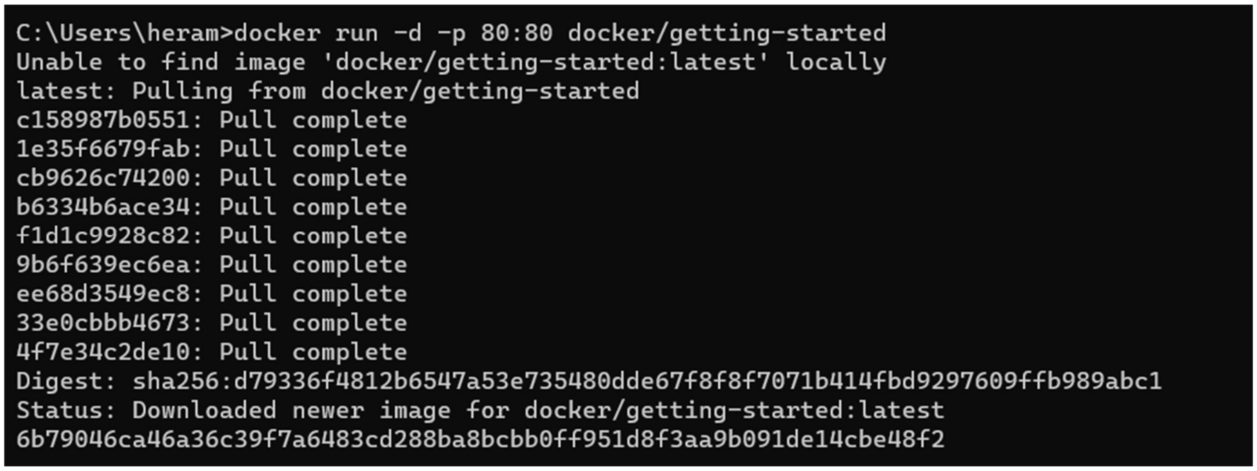
Exploring Docker Images:



```
C:\Windows\System32>docker images
REPOSITORY               TAG       IMAGE ID       CREATED         SIZE
nginx                    latest    904b8cb13b93   2 weeks ago     142MB
redis                    latest    f9c173b0f012   2 weeks ago     117MB
docker/getting-started   latest    3e4394f6b72f   2 months ago    47MB
```

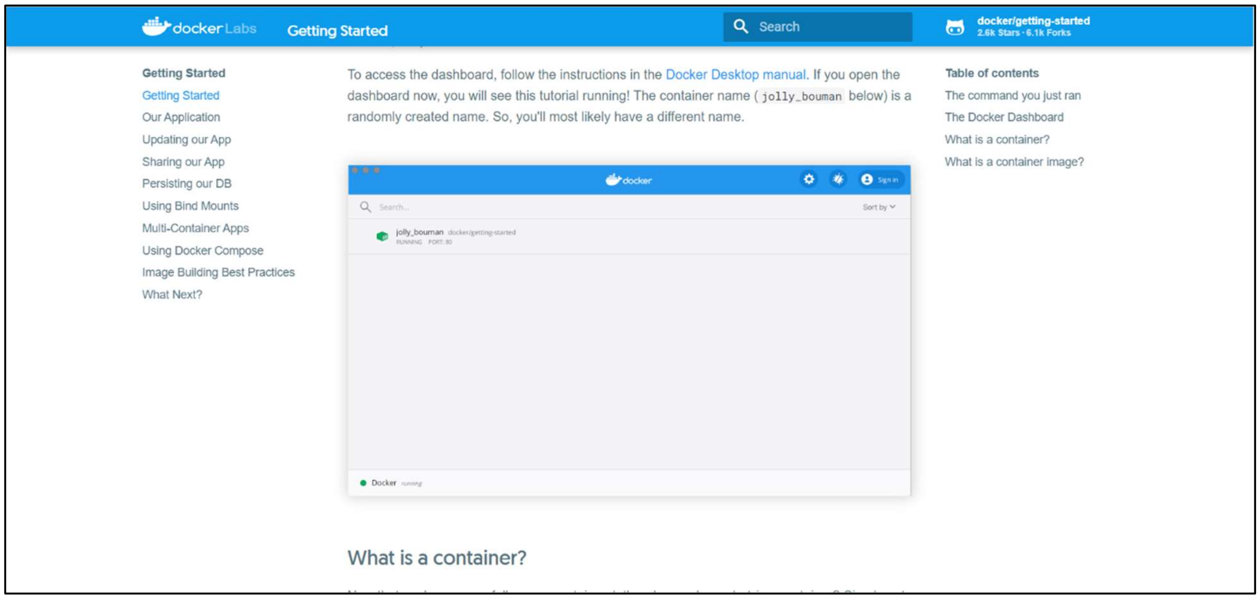
Exploring Docker Containers:

a. Starting A Container from Image:



```
C:\Users\heram>docker run -d -p 80:80 docker/getting-started
Unable to find image 'docker/getting-started:latest' locally
latest: Pulling from docker/getting-started
c158987b0551: Pull complete
1e35f6679fab: Pull complete
cb9626c74200: Pull complete
b6334b6ace34: Pull complete
f1d1c9928c82: Pull complete
9b6f639ec6ea: Pull complete
ee68d3549ec8: Pull complete
33e0cbbb4673: Pull complete
4f7e34c2de10: Pull complete
Digest: sha256:d79336f4812b6547a53e735480dde67f8f8f7071b414fbd9297609ffb989abc1
Status: Downloaded newer image for docker/getting-started:latest
6b79046ca46a36c39f7a6483cd288ba8bcbb0ff951d8f3aa9b091de14cbe48f2
```

Output At Localhost:80



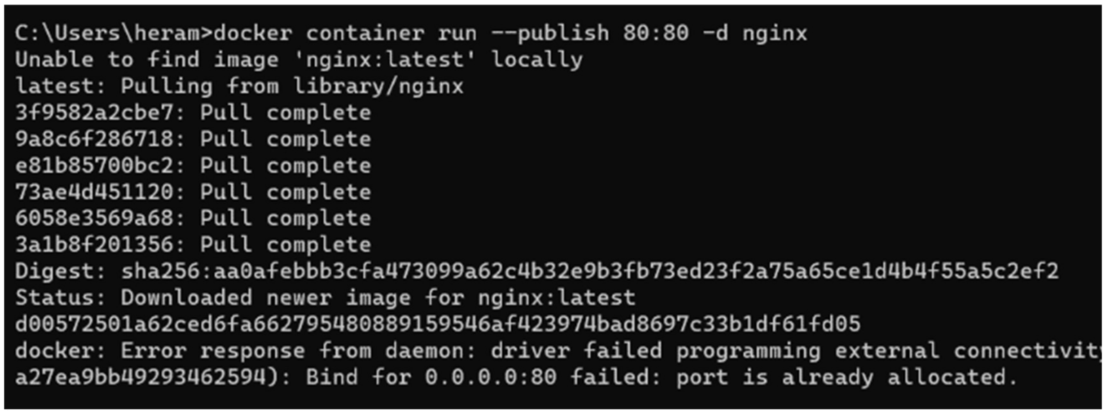
Getting Started
Getting Started
Our Application
Updating our App
Sharing our App
Persisting our DB
Using Bind Mounts
Multi-Container Apps
Using Docker Compose
Image Building Best Practices
What Next?

To access the dashboard, follow the instructions in the Docker Desktop manual. If you open the dashboard now, you will see this tutorial running! The container name (`jolly_bouman` below) is a randomly created name. So, you'll most likely have a different name.

What is a container?

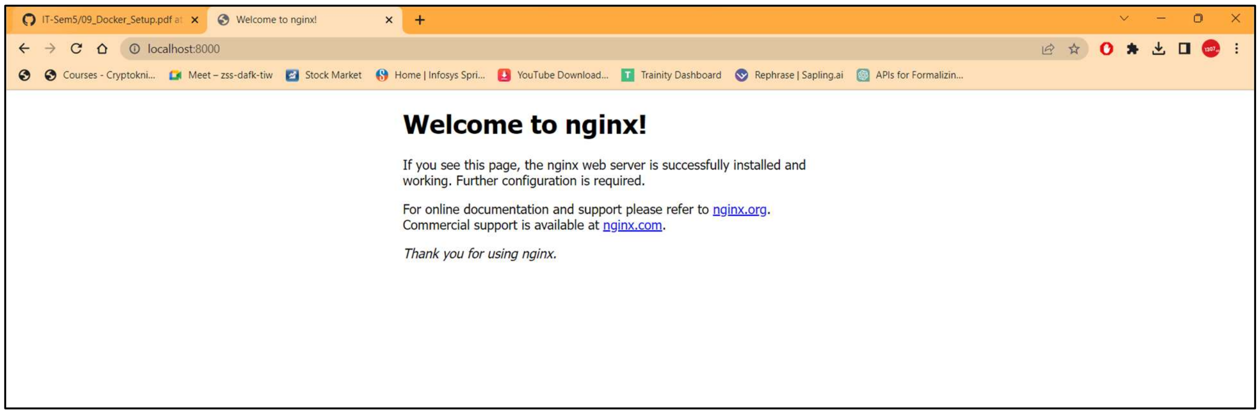
Table of contents
The command you just ran
The Docker Dashboard
What is a container?
What is a container image?

B. Starting Another Container from Image:



```
C:\Users\heram>docker container run --publish 80:80 -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
3f9582a2cbe7: Pull complete
9a8c6f286718: Pull complete
e81b85700bc2: Pull complete
73ae4d451120: Pull complete
6058e3569a68: Pull complete
3a1b8f201356: Pull complete
Digest: sha256:aa0afebbb3cfa473099a62c4b32e9b3fb73ed23f2a75a65ce1d4b4f55a5c2ef2
Status: Downloaded newer image for nginx:latest
d00572501a62ced6fa6627954808891595 46af423974bad8697c33b1df61fd05
docker: Error response from daemon: driver failed programming external connectivit
a27ea9bb49293462594): Bind for 0.0.0.0:80 failed: port is already allocated.
```

Output At Localhost:8000



Welcome to nginx!

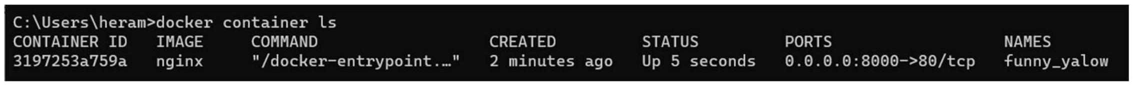
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

Thank you for using nginx.

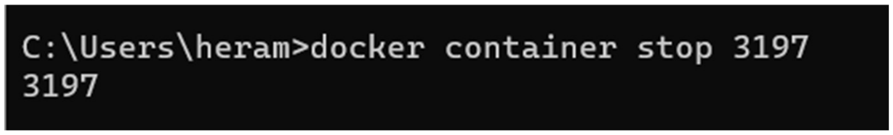
Exploring Container Commands:

1. **Listing Out Running Containers:**



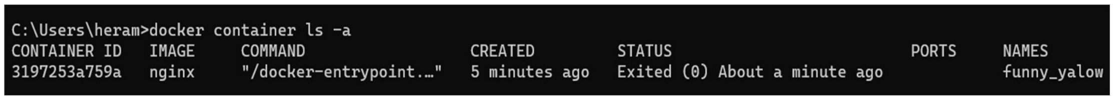
```
C:\Users\heram>docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                  NAMES
3197253a759a   nginx     "/docker-entrypoint.…"   2 minutes ago   Up 5 seconds   0.0.0.0:8000->80/tcp   funny_yalow
```

2. **Stopping a running container**



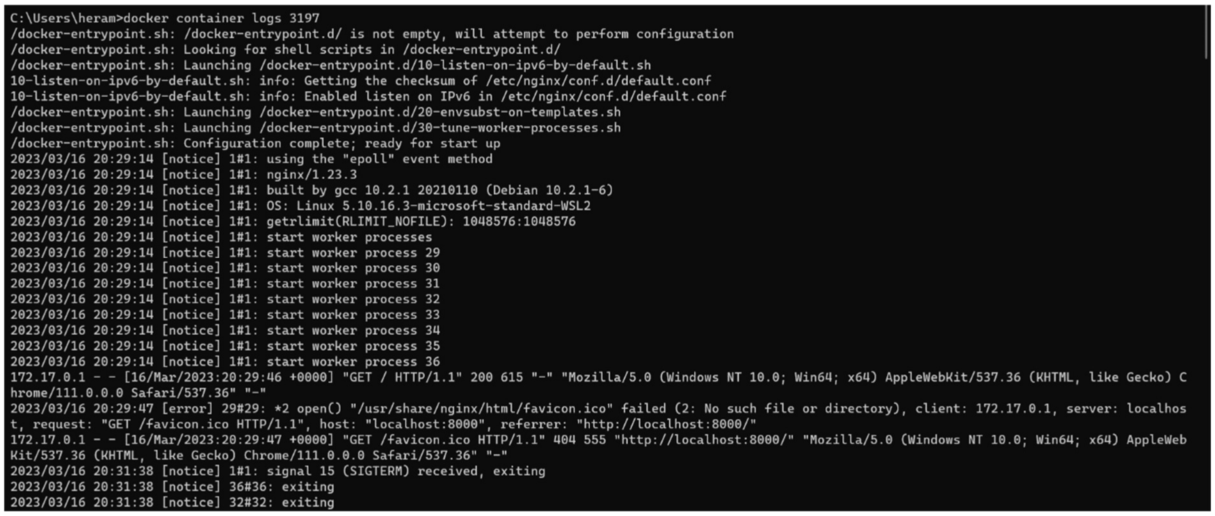
```
C:\Users\heram>docker container stop 3197
3197
```

3. **Listing Out Containers that are running and stopped.**



```
C:\Users\heram>docker container ls -a
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS                          PORTS     NAMES
3197253a759a   nginx     "/docker-entrypoint.…"   5 minutes ago   Exited (0) About a minute ago             funny_yalow
```

4. **Exploring Specific Logs Of a Container:**



```
C:\Users\heram>docker container logs 3197
/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
2023/03/16 20:29:14 [notice] 1#1: using the "epoll" event method
2023/03/16 20:29:14 [notice] 1#1: nginx/1.23.3
2023/03/16 20:29:14 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/03/16 20:29:14 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2023/03/16 20:29:14 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/03/16 20:29:14 [notice] 1#1: start worker processes
2023/03/16 20:29:14 [notice] 1#1: start worker process 29
2023/03/16 20:29:14 [notice] 1#1: start worker process 30
2023/03/16 20:29:14 [notice] 1#1: start worker process 31
2023/03/16 20:29:14 [notice] 1#1: start worker process 32
2023/03/16 20:29:14 [notice] 1#1: start worker process 33
2023/03/16 20:29:14 [notice] 1#1: start worker process 34
2023/03/16 20:29:14 [notice] 1#1: start worker process 35
2023/03/16 20:29:14 [notice] 1#1: start worker process 36
172.17.0.1 - - [16/Mar/2023:20:29:46 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
hrome/111.0.0.0 Safari/537.36" "-"
2023/03/16 20:29:47 [error] 29#29: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhos
t, request: "GET /favicon.ico HTTP/1.1", host: "localhost:8000", referrer: "http://localhost:8000/"
172.17.0.1 - - [16/Mar/2023:20:29:47 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost:8000/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWeb
Kit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/537.36" "-"
2023/03/16 20:31:38 [notice] 1#1: signal 15 (SIGTERM) received, exiting
2023/03/16 20:31:38 [notice] 36#36: exiting
2023/03/16 20:31:38 [notice] 32#32: exiting
```