Experiment 09 - Docker Architecture

Roll No.	37
Name	Mikil Lalwani
Class	D15-B
Subject	DevOps Lab
LO Mapped	LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements. LO5: To understand the concept of containerization and Analyze the Containerization of OS images and deployment of applications over Docker.

Aim:

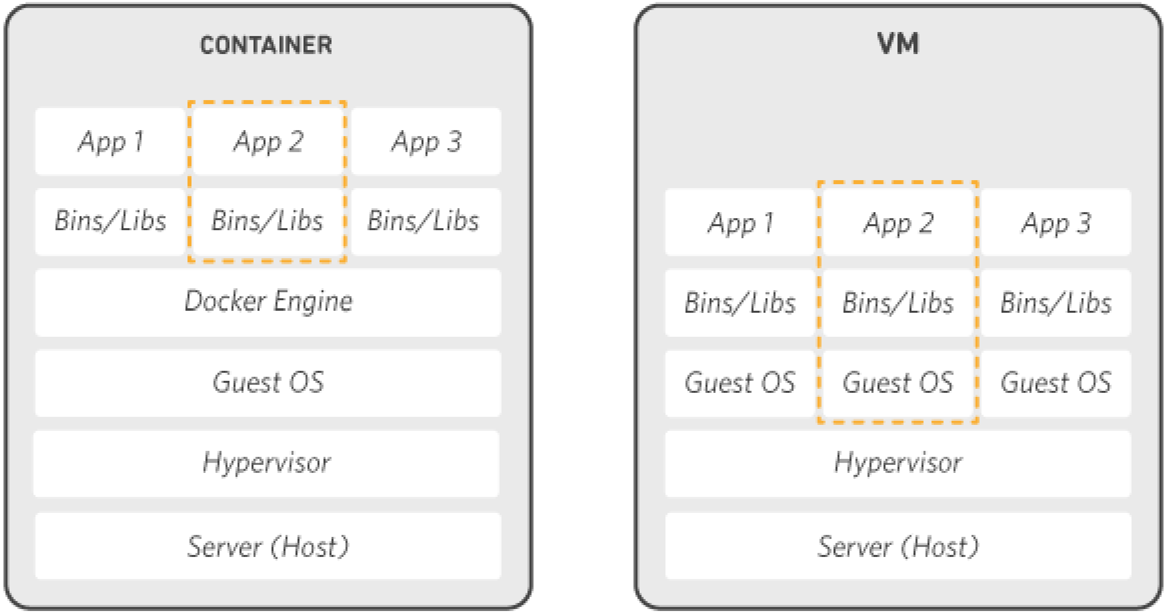
To understand Docker Architecture and Container Life Cycle, install Docker, and execute docker commands to manage images and interact with containers.

Introduction:

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run. Running Docker on AWS provides developers and admins with a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.

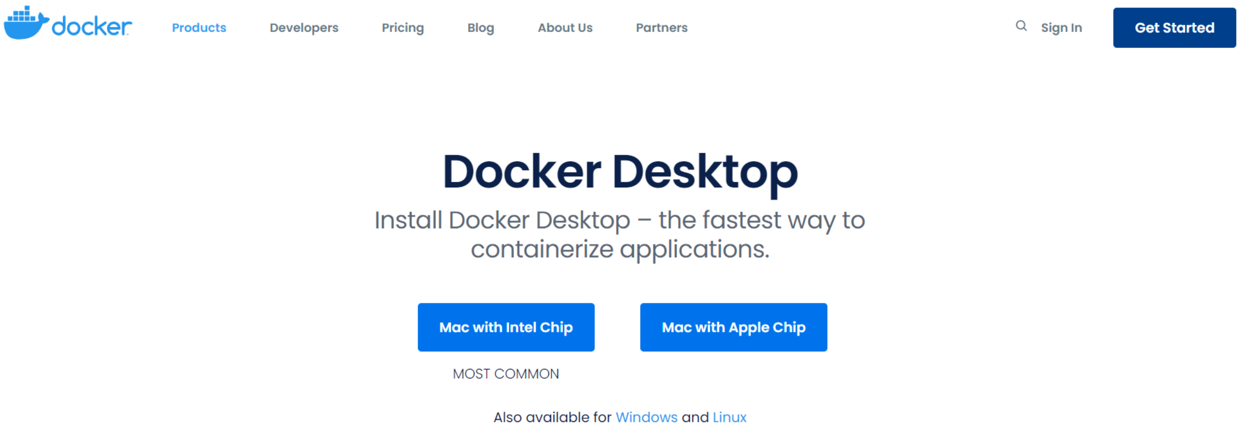
How Docker works -

Docker works by providing a standard way to run your code. Docker is an operating system for containers. Similar to how a virtual machine virtualizes (removes the need to directly manage) server hardware, containers virtualize the operating system of a server. Docker is installed on each server and provides simple commands you can use to build, start, or stop containers.



Installation:

1. Open up the Docker website and click on the Download button to download Docker Desktop for your Operating System.



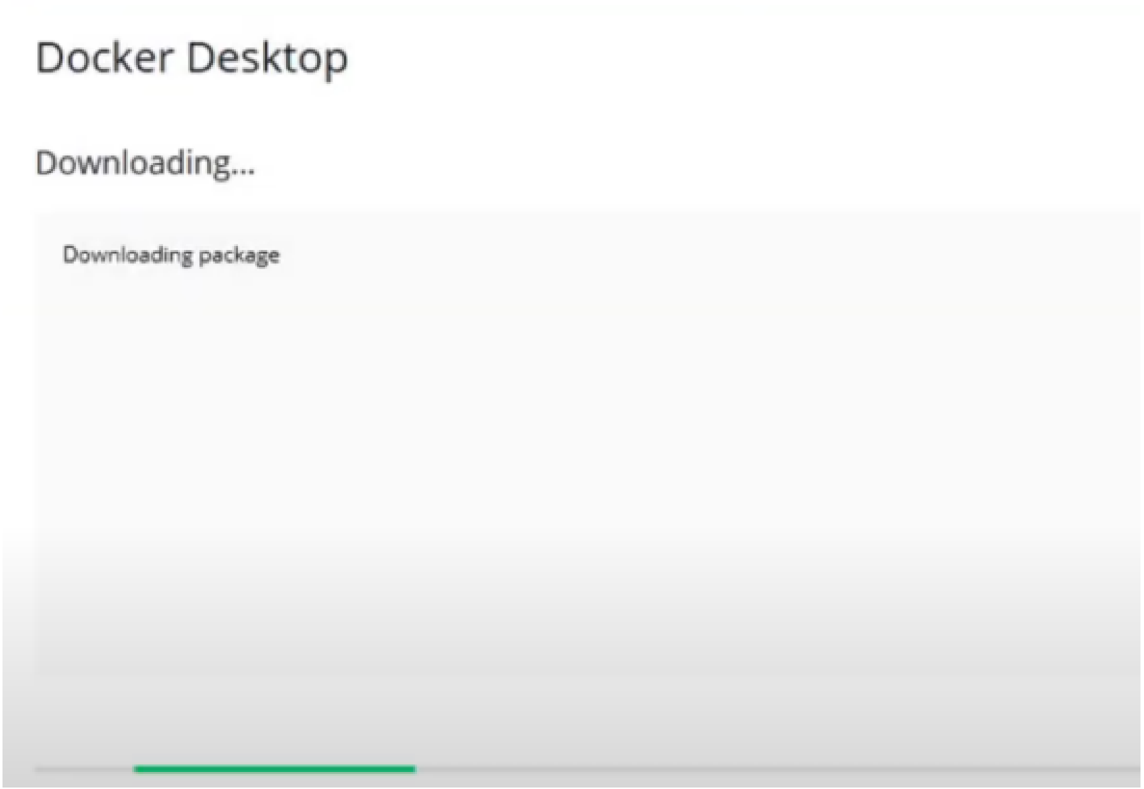
Docker Desktop

Install Docker Desktop – the fastest way to containerize applications.

MOST COMMON

Also available for Windows and Linux

2. Open the Installer and wait for Docker to download its prerequisites.

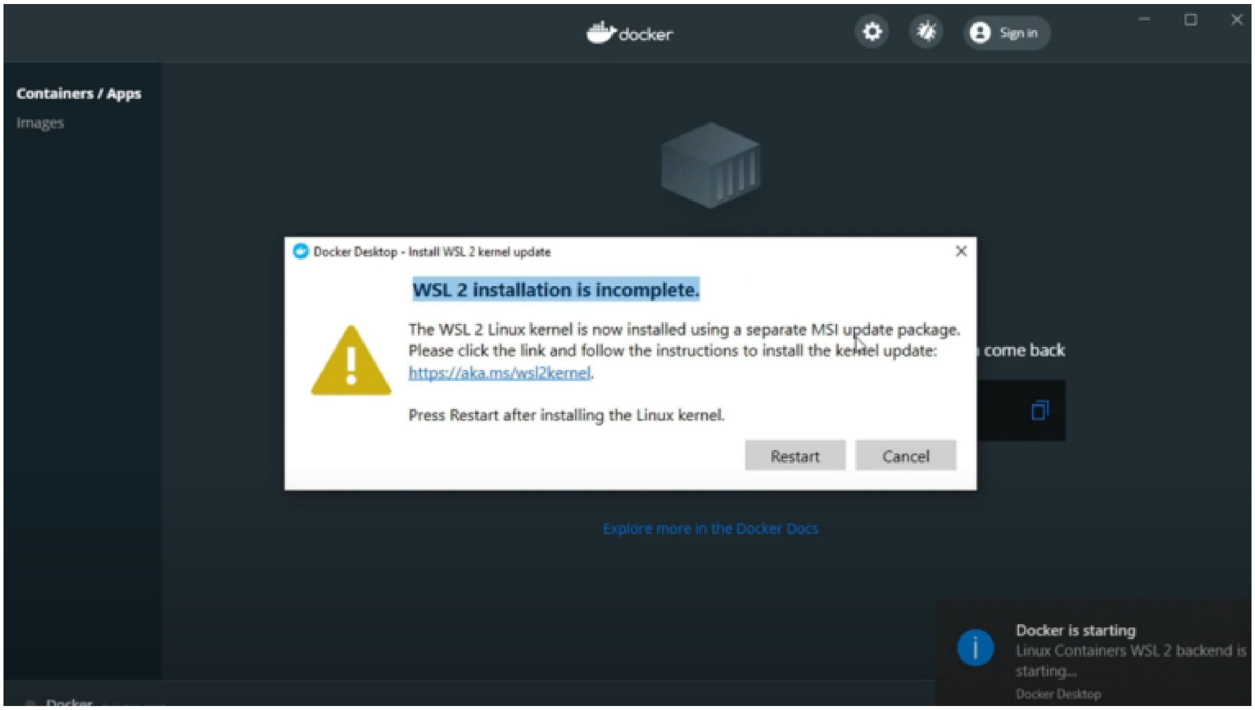


Docker Desktop

Downloading...

Downloading package

3. Restart your machine once the installation is complete.
4. Start Docker Desktop.



Docker Desktop - Install WSL 2 kernel update

WSL 2 installation is incomplete.

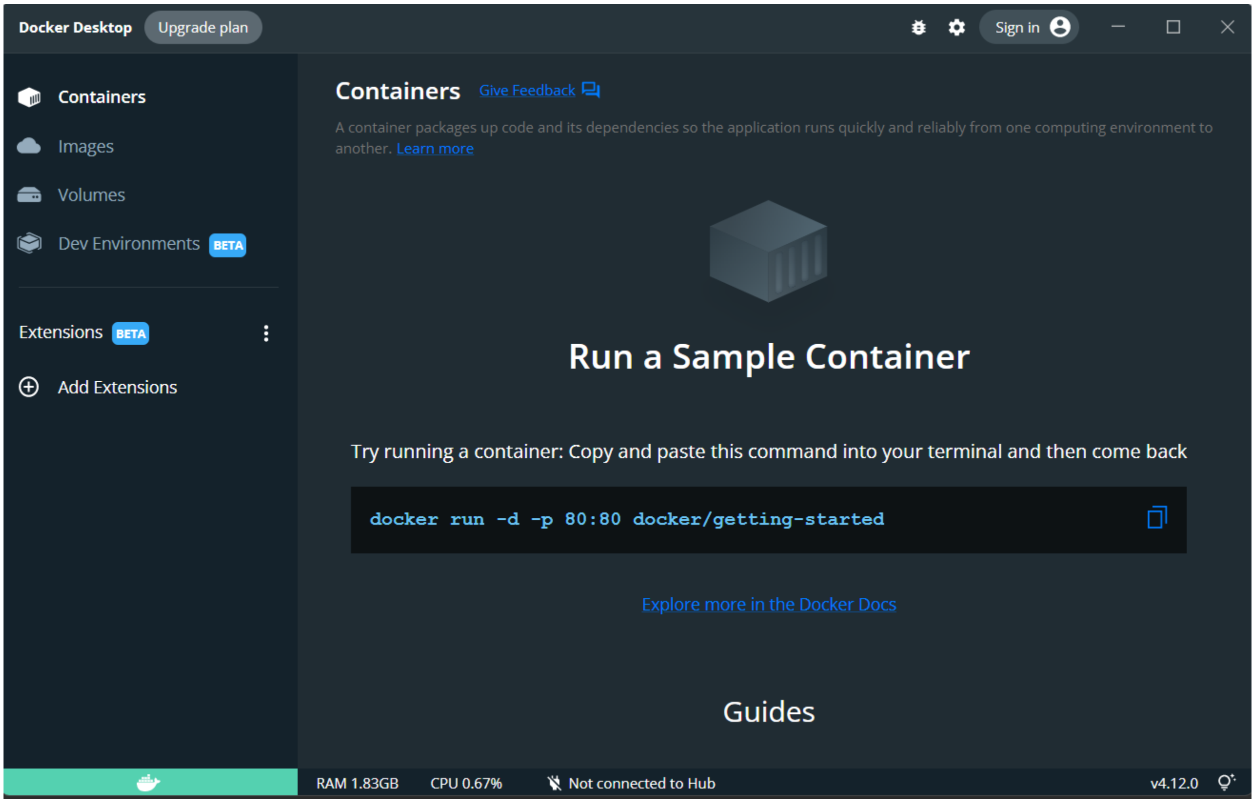
The WSL 2 Linux kernel is now installed using a separate MSI update package. Please click the link and follow the instructions to install the kernel update: https://aka.ms/wsl2kernel.

Press Restart after installing the Linux kernel.

Docker is starting
Linux Containers WSL 2 backend is starting...
Docker Desktop

Download the required WSL2 kernel and restart your machine to complete the installation if prompted.

5. With that, Docker Desktop is ready.



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

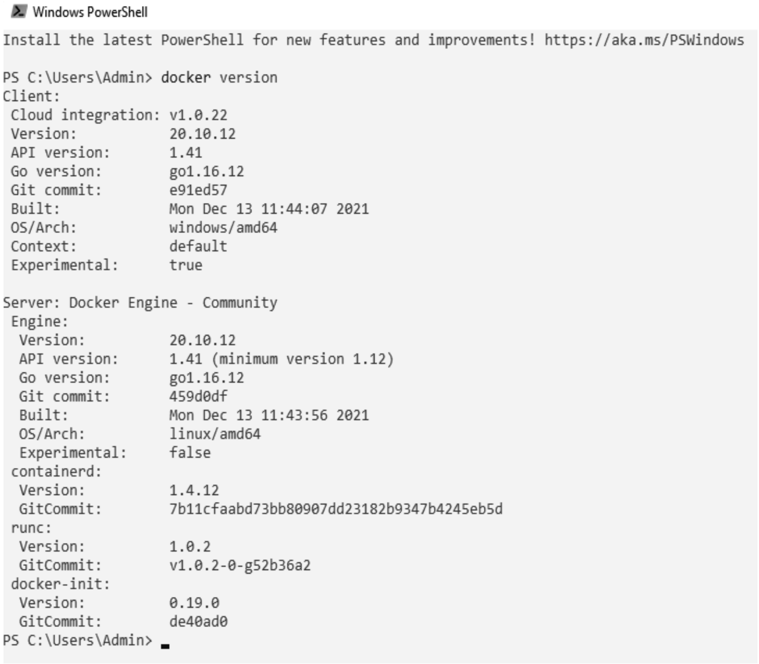
Run a Sample Container

Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

Docker Commands

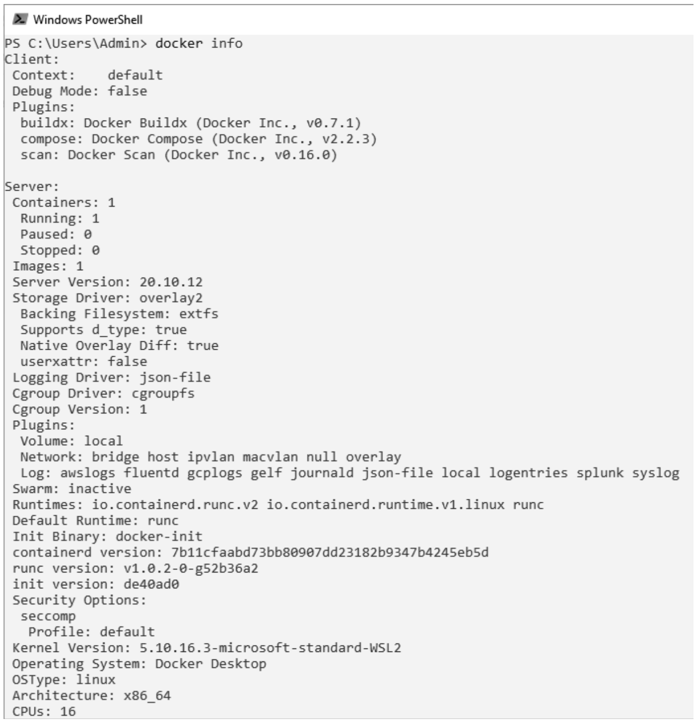
1. Checking Docker Version
   docker version
   docker info



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> docker version
Client:
 Cloud integration: v1.0.22
 Version:           20.10.12
 API version:       1.41
 Go version:        go1.16.12
 Git commit:        e91ed57
 Built:             Mon Dec 13 11:44:07 2021
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Engine - Community
 Engine:
  Version:          20.10.12
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.16.12
  Git commit:       459d0df
  Built:            Mon Dec 13 11:43:56 2021
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.4.12
  GitCommit:        7b11cfaabd73bb80907dd23182b9347b4245eb5d
 runc:
  Version:          1.0.2
  GitCommit:        v1.0.2-0-g52b36a2
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
PS C:\Users\Admin>
```



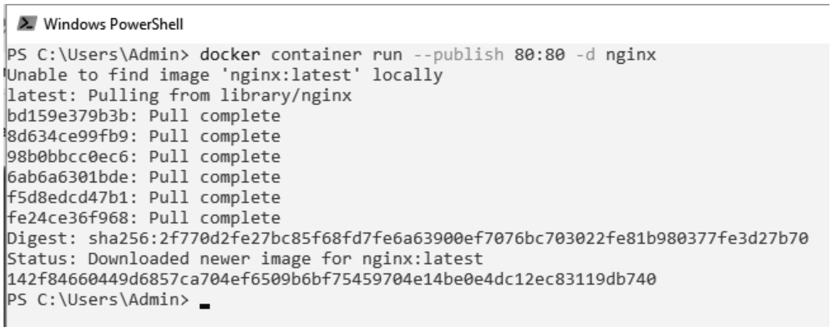
```
PS C:\Users\Admin> docker info
Client:
 Context:    default
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc., v0.7.1)
  compose: Docker Compose (Docker Inc., v2.2.3)
  scan: Docker Scan (Docker Inc., v0.16.0)

Server:
 Containers: 1
  Running: 1
  Paused: 0
  Stopped: 0
 Images: 1
 Server Version: 20.10.12
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7b11cfaabd73bb80907dd23182b9347b4245eb5d
 runc version: v1.0.2-0-g52b36a2
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.10.16.3-microsoft-standard-WSL2
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 16
```

2. Starting a Container from an Image

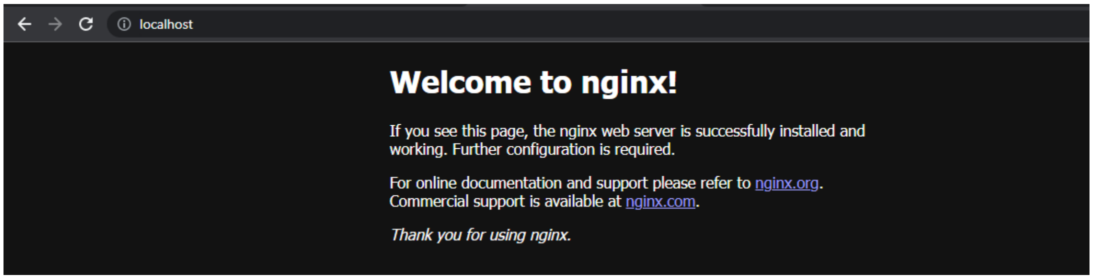
We can run Docker Containers from a pre-existing image or docker will pull the specified image from the Docker hub.

For this example, we will run an Nginx server in a docker container on port 80. docker container run --publish 80:80 -d nginx



```
PS C:\Users\Admin> docker container run --publish 80:80 -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bd159e379b3b: Pull complete
8d634ce99fb9: Pull complete
98b0bbcc0ec6: Pull complete
6ab6a6301bde: Pull complete
f5d8edcd47b1: Pull complete
fe24ce36f968: Pull complete
Digest: sha256:2f770d2fe27bc85f68fd7fe6a63900ef7076bc703022fe81b980377fe3d27b70
Status: Downloaded newer image for nginx:latest
142f84660449d6857ca704ef6509b6bf75459704e14be0e4dc12ec83119db740
PS C:\Users\Admin> _
```

On your browser, open up localhost:80 and check to see if the nginx server is up.



Welcome to nginx!

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

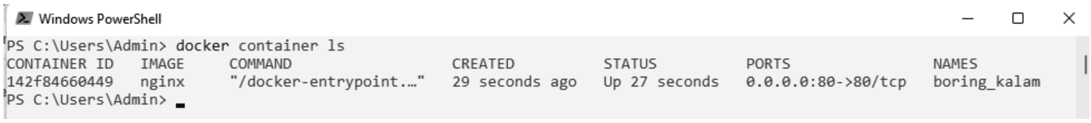
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

3. Listing out Containers

   We can find a list of running containers on our machine.

   docker container ls

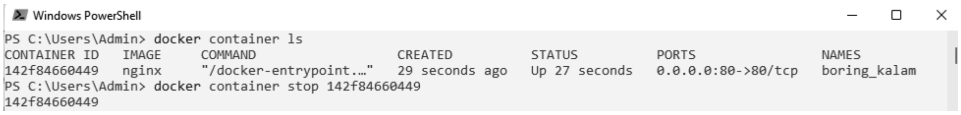


```
PS C:\Users\Admin> docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                NAMES
142f84660449   nginx     "/docker-entrypoint.…"   29 seconds ago   Up 27 seconds   0.0.0.0:80->80/tcp   boring_kalam
PS C:\Users\Admin> _
```

4. Stopping a Container

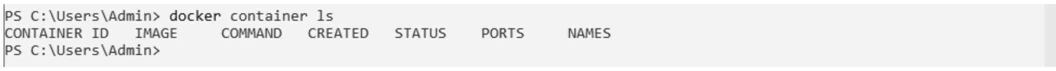
   We can stop a running container using the docker container stop command by providing the id we found above. We only need to provide the initial few letters of the id, until it's totally unique.

   docker container stop



```
PS C:\Users\Admin> docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                NAMES
142f84660449   nginx     "/docker-entrypoint.…"   29 seconds ago   Up 27 seconds   0.0.0.0:80->80/tcp   boring_kalam
PS C:\Users\Admin> docker container stop 142f84660449
142f84660449
```

If we now run docker container ls we get an empty response.

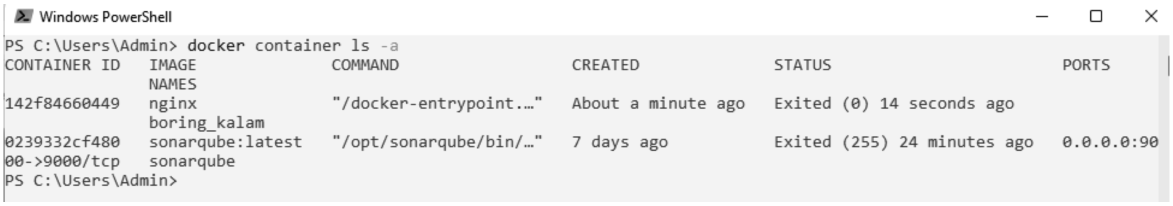


```
PS C:\Users\Admin> docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\Admin>
```

5. Listing out All Containers

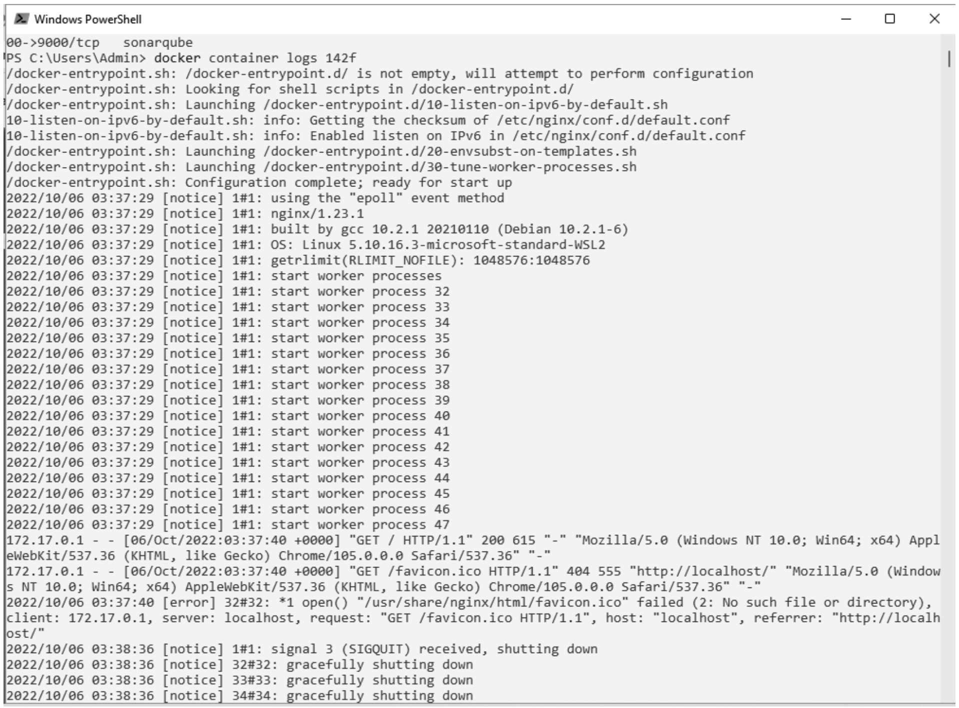
   We can use the -a flag to the previous list command to find a list of all containers, even those which have stopped.

   docker container ls -a



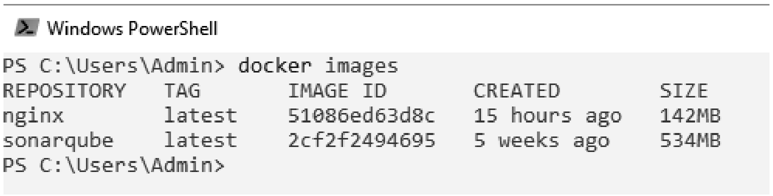
```
PS C:\Users\Admin> docker container ls -a
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS                      PORTS
               NAMES
142f84660449   nginx              "/docker-entrypoint.…"   About a minute ago   Exited (0) 14 seconds ago
               boring_kalam
0239332cf480   sonarqube:latest   "/opt/sonarqube/bin/…"   7 days ago           Exited (255) 24 minutes ago   0.0.0.0:90
00->9000/tcp   sonarqube
PS C:\Users\Admin>
```

6. Show container logs
   We can use the command logs to show logs for a specified container -
   docker container logs <id>



```
00->9000/tcp   sonarqube
PS C:\Users\Admin> docker container logs 142f
/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
2022/10/06 03:37:29 [notice] 1#1: using the "epoll" event method
2022/10/06 03:37:29 [notice] 1#1: nginx/1.23.1
2022/10/06 03:37:29 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/10/06 03:37:29 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2022/10/06 03:37:29 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/10/06 03:37:29 [notice] 1#1: start worker processes
2022/10/06 03:37:29 [notice] 1#1: start worker process 32
2022/10/06 03:37:29 [notice] 1#1: start worker process 33
2022/10/06 03:37:29 [notice] 1#1: start worker process 34
2022/10/06 03:37:29 [notice] 1#1: start worker process 35
2022/10/06 03:37:29 [notice] 1#1: start worker process 36
2022/10/06 03:37:29 [notice] 1#1: start worker process 37
2022/10/06 03:37:29 [notice] 1#1: start worker process 38
2022/10/06 03:37:29 [notice] 1#1: start worker process 39
2022/10/06 03:37:29 [notice] 1#1: start worker process 40
2022/10/06 03:37:29 [notice] 1#1: start worker process 41
2022/10/06 03:37:29 [notice] 1#1: start worker process 42
2022/10/06 03:37:29 [notice] 1#1: start worker process 43
2022/10/06 03:37:29 [notice] 1#1: start worker process 44
2022/10/06 03:37:29 [notice] 1#1: start worker process 45
2022/10/06 03:37:29 [notice] 1#1: start worker process 46
2022/10/06 03:37:29 [notice] 1#1: start worker process 47
172.17.0.1 - - [06/Oct/2022:03:37:40 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl
eWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
172.17.0.1 - - [06/Oct/2022:03:37:40 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Window
s NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
2022/10/06 03:37:40 [error] 32#32: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory),
client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referrer: "http://localh
ost/"
2022/10/06 03:38:36 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2022/10/06 03:38:36 [notice] 32#32: gracefully shutting down
2022/10/06 03:38:36 [notice] 33#33: gracefully shutting down
2022/10/06 03:38:36 [notice] 34#34: gracefully shutting down
```

7. Listing out Images
   We can use the docker images command to show a list of docker images we locally have.
   docker images



```
PS C:\Users\Admin> docker images
REPOSITORY   TAG      IMAGE ID       CREATED         SIZE
nginx        latest   51086ed63d8c   15 hours ago    142MB
sonarqube    latest   2cf2f2494695   5 weeks ago     534MB
PS C:\Users\Admin>
```

<u>Conclusion</u>:

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.