**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.

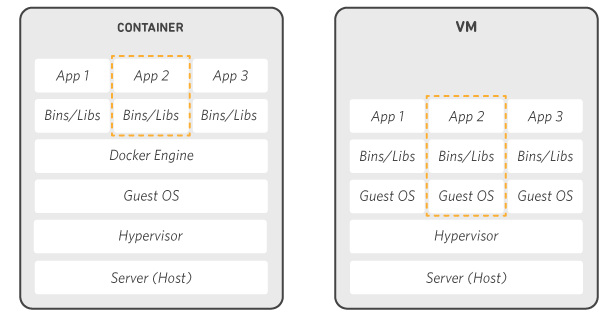
**Introduction**:

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.

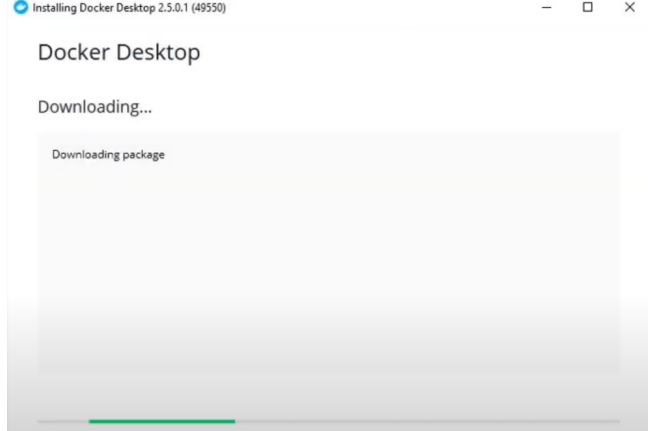


**Installation:**

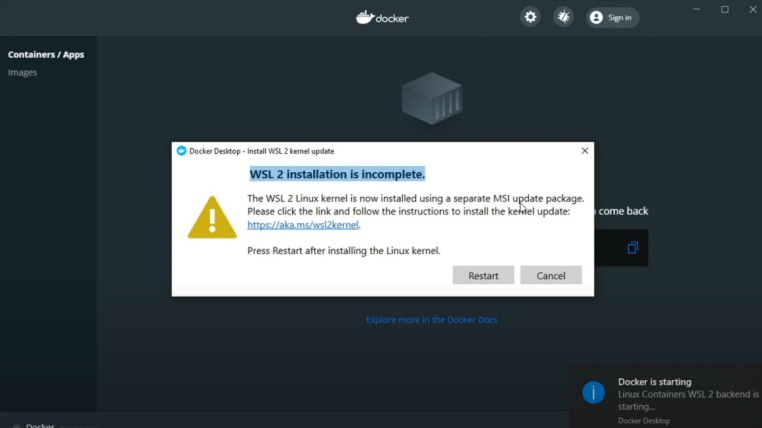
1. Open up the Docker website and click on the Download button to download Docker Desktop for your Operating System.

****

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

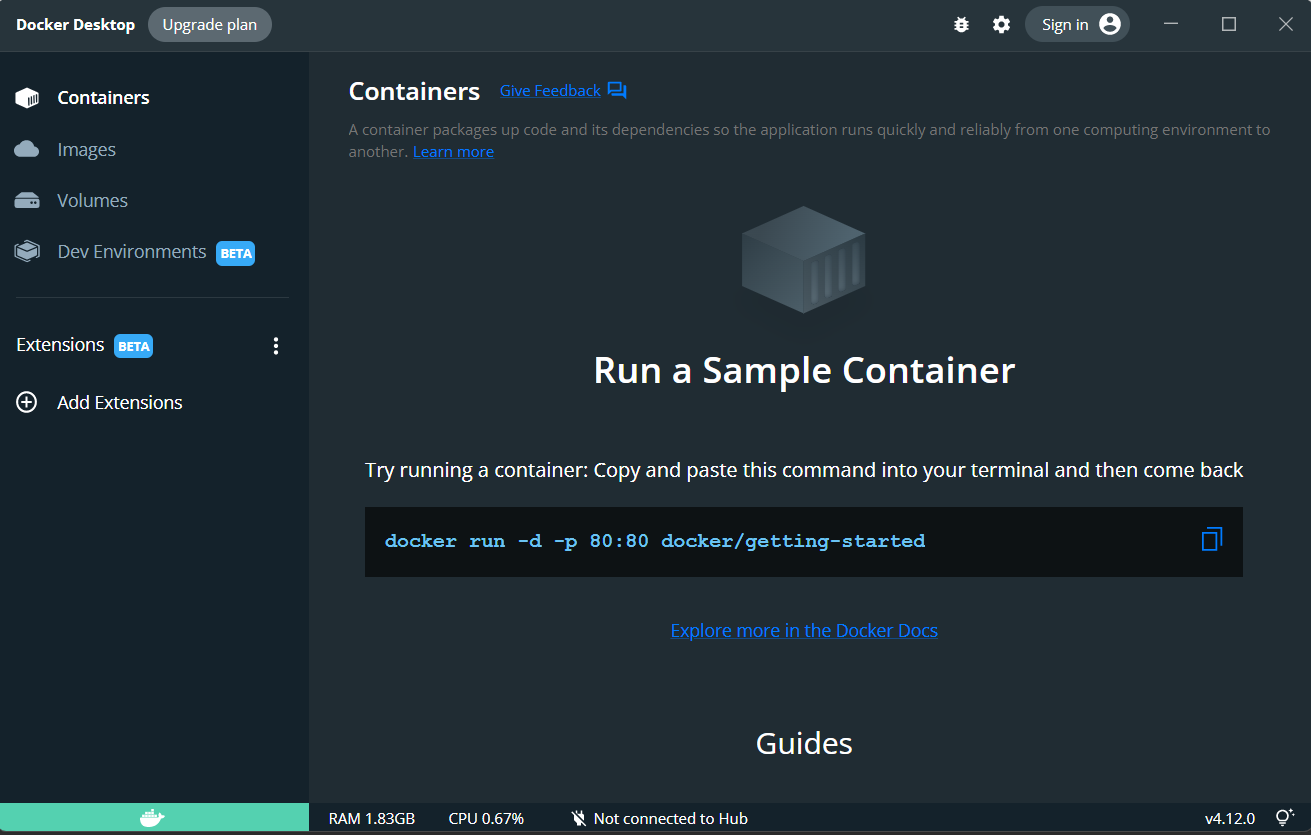


1. Restart your machine once the installation is complete.
2. Start Docker Desktop.



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

1. With that, Docker Desktop is ready.

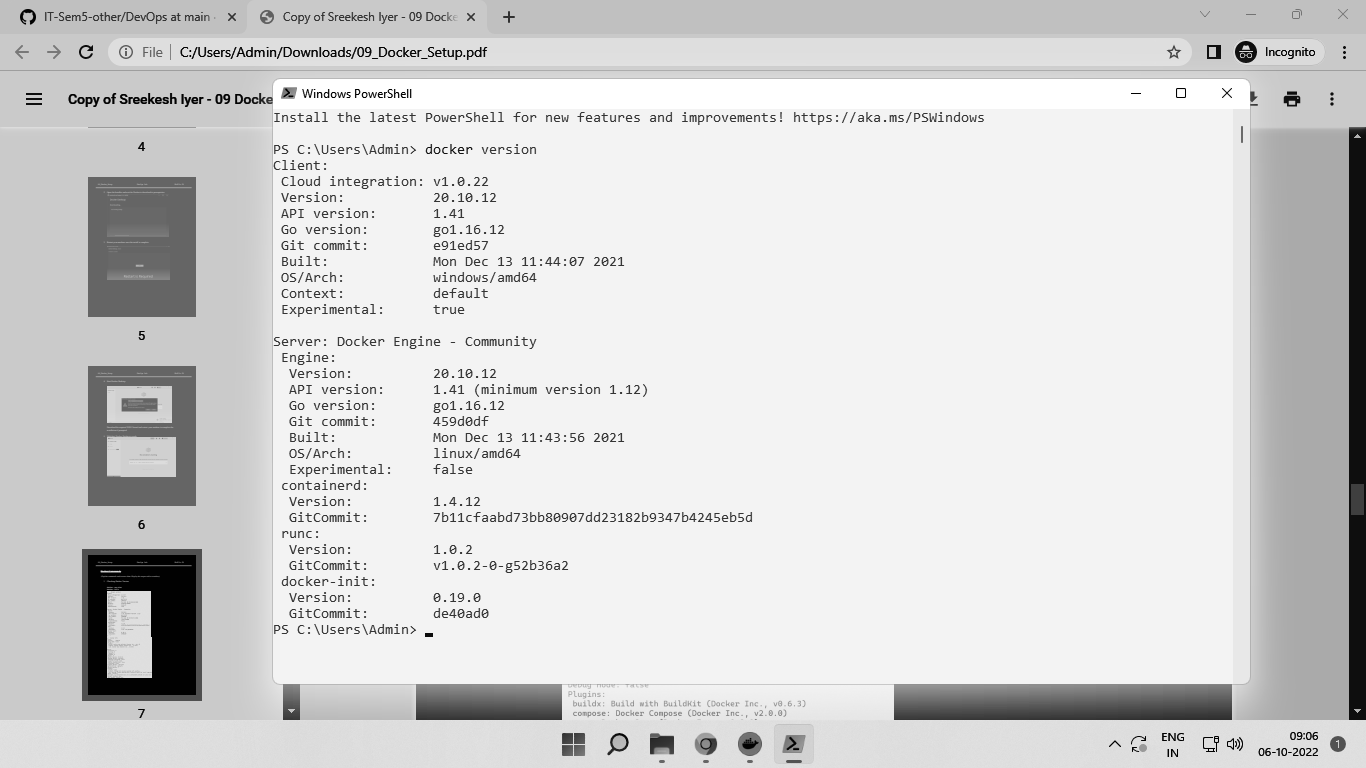


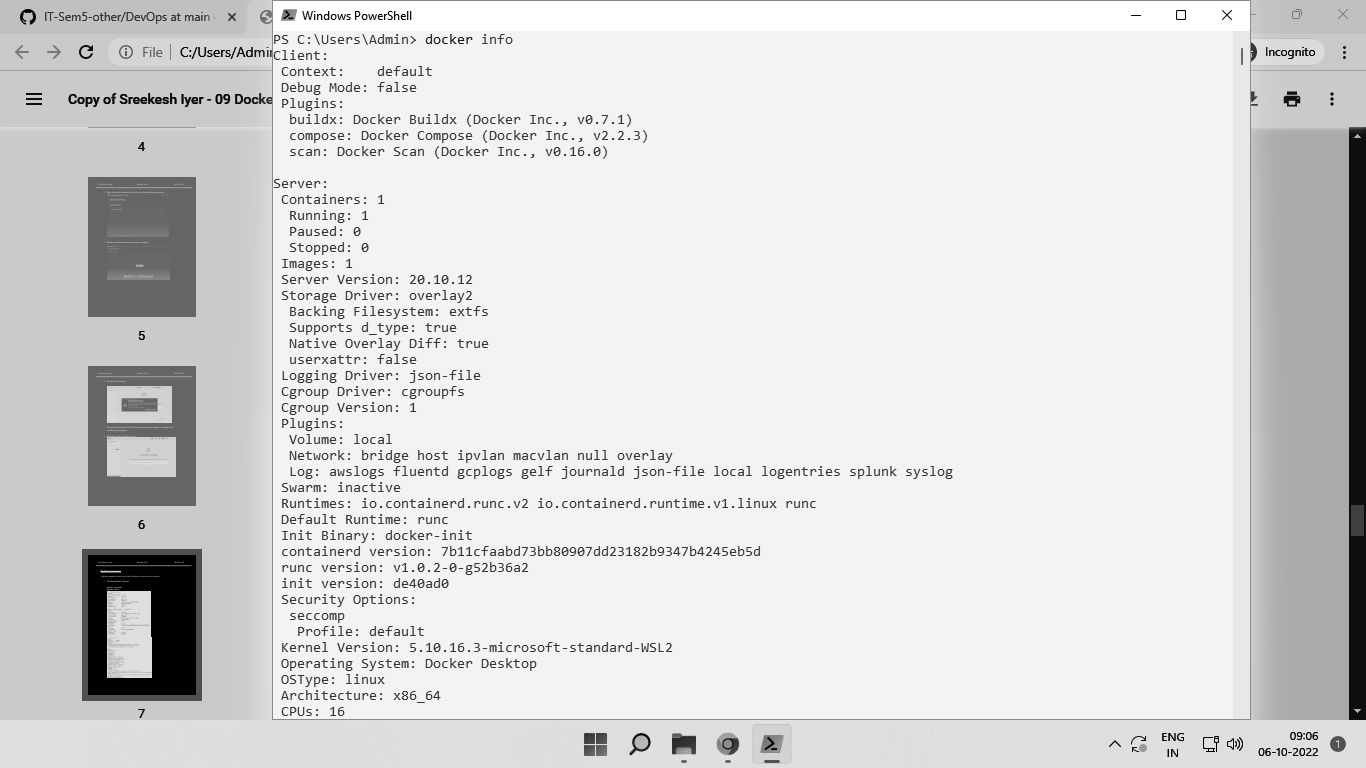
**Docker Commands**

1. Checking Docker Version

docker version

docker info

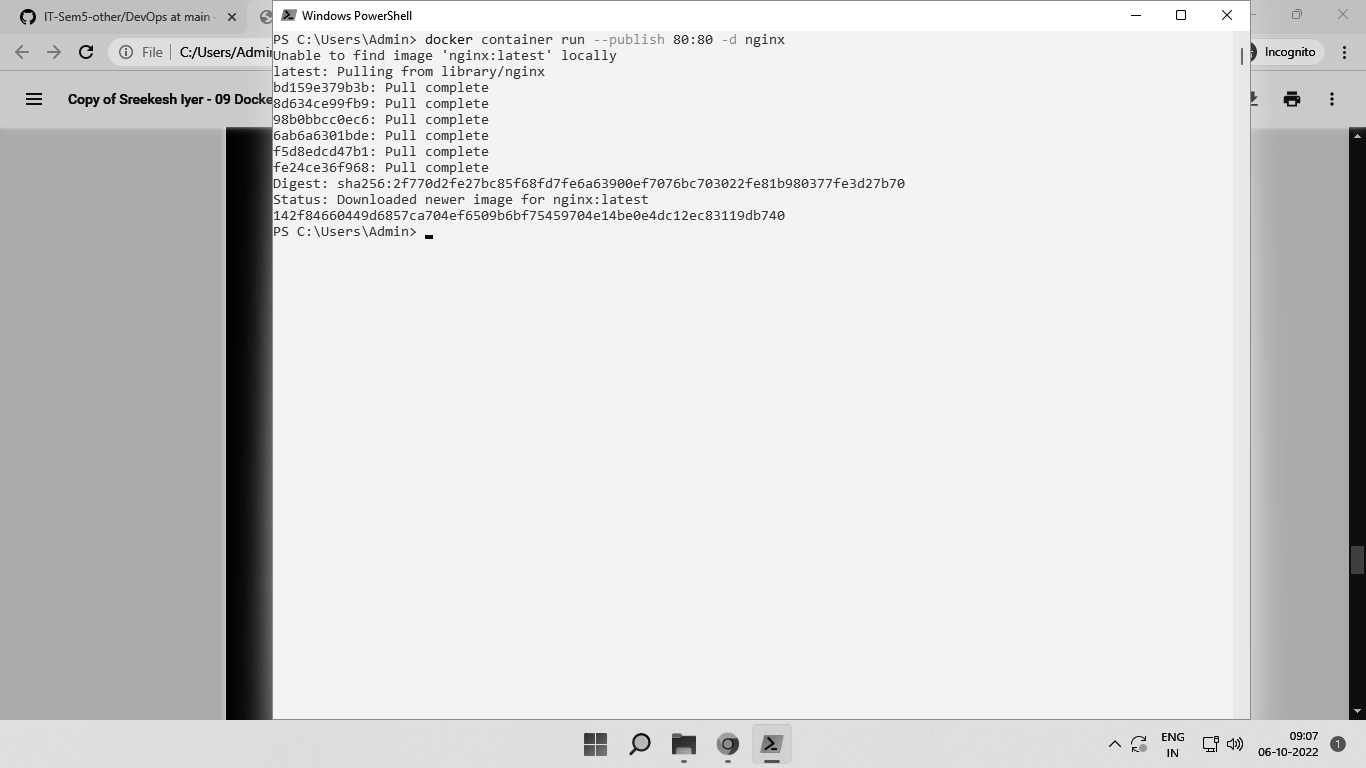




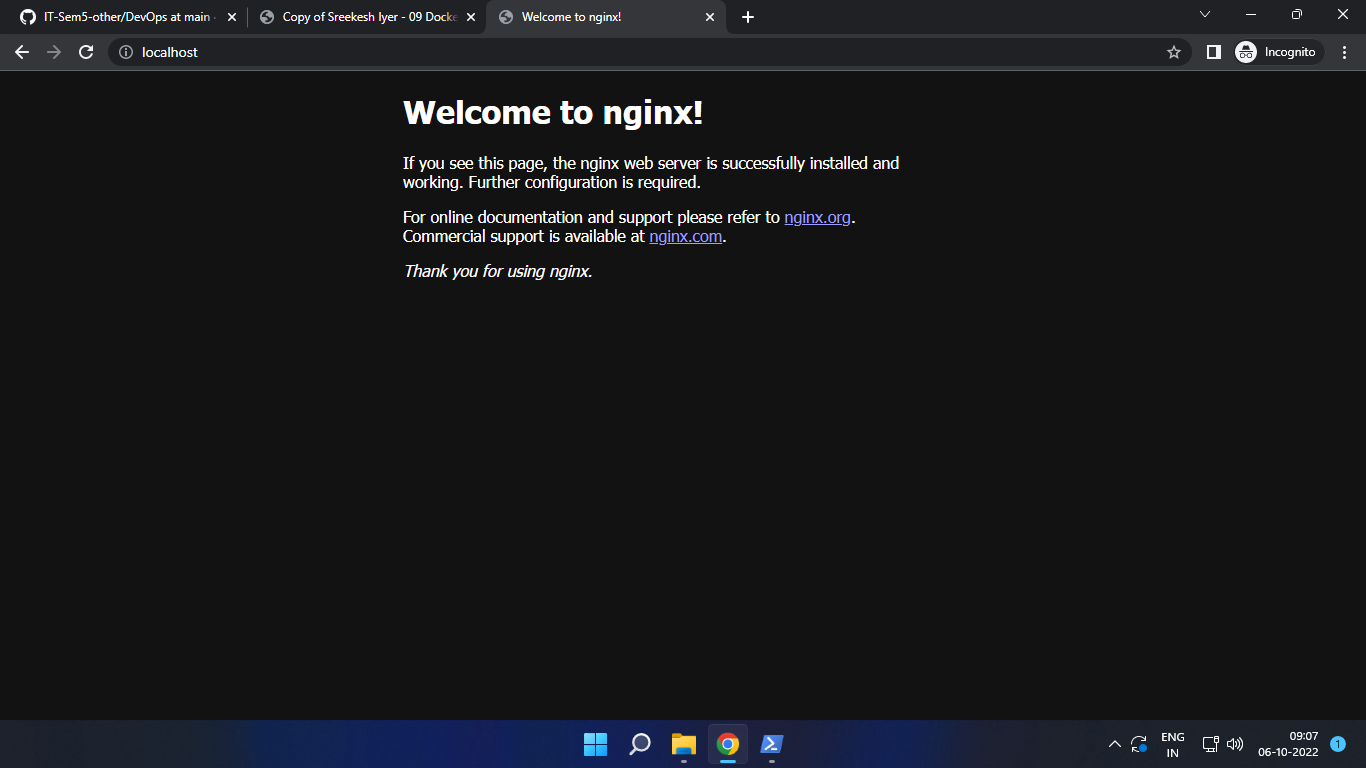
1. Starting a Container from an Image

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



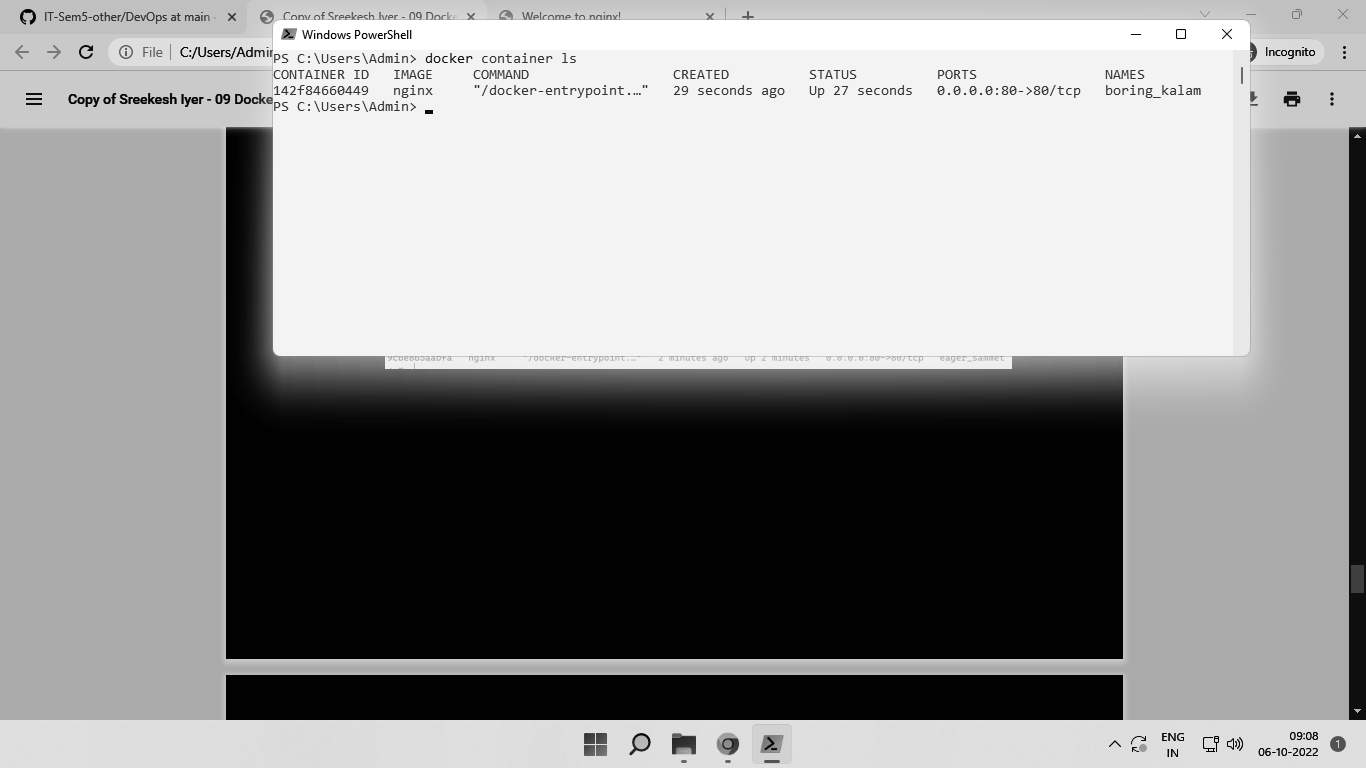
On your browser, open up localhost:80 and check to see if the nginx server is up.



1. Listing out Containers

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

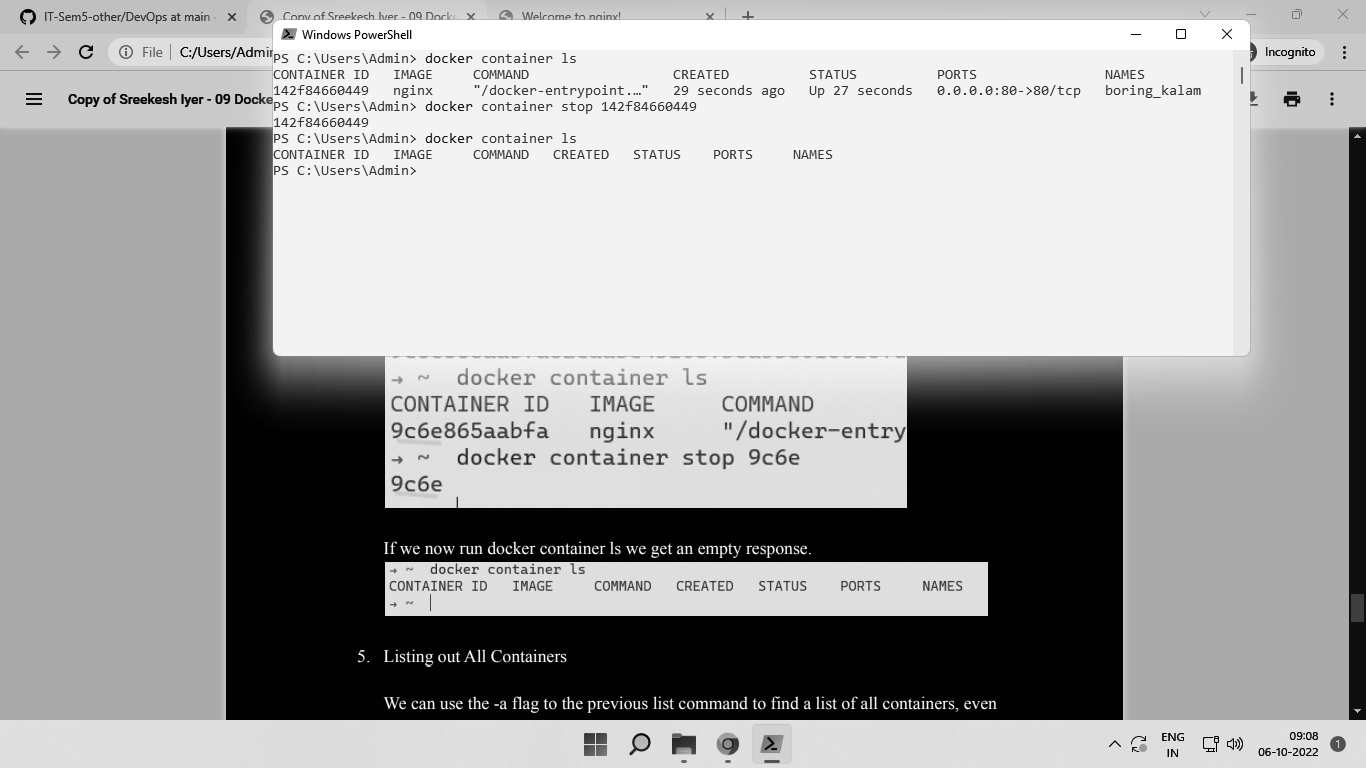
docker container ls



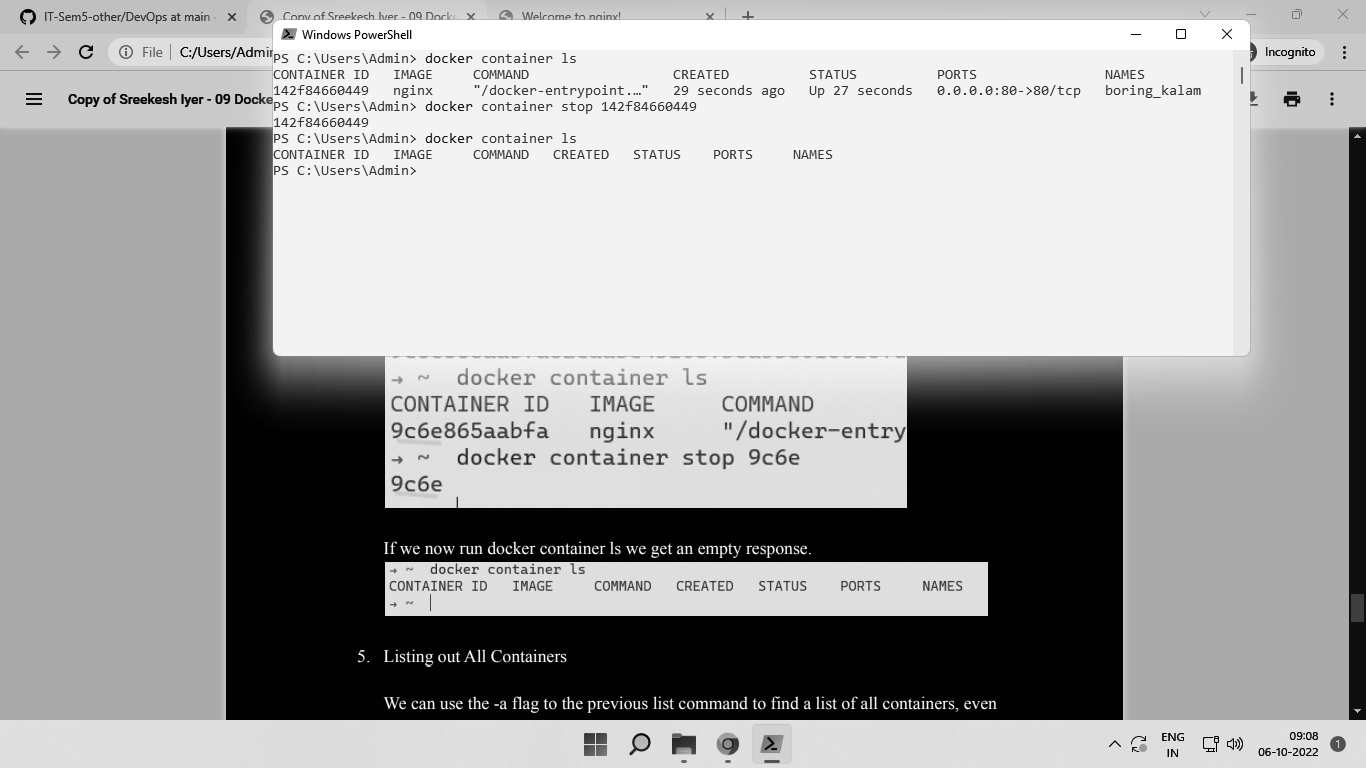
1. Stopping a Container

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



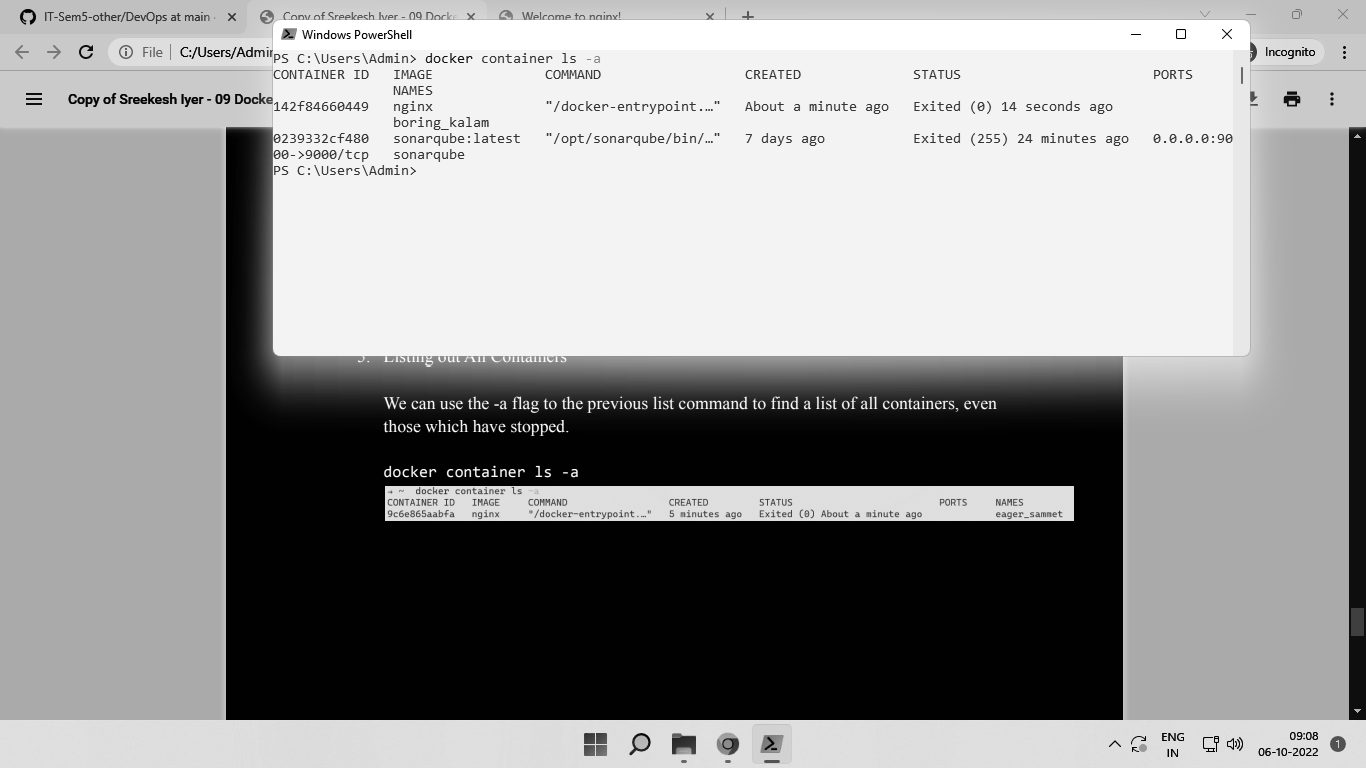
If we now run docker container ls we get an empty response.



1. 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.

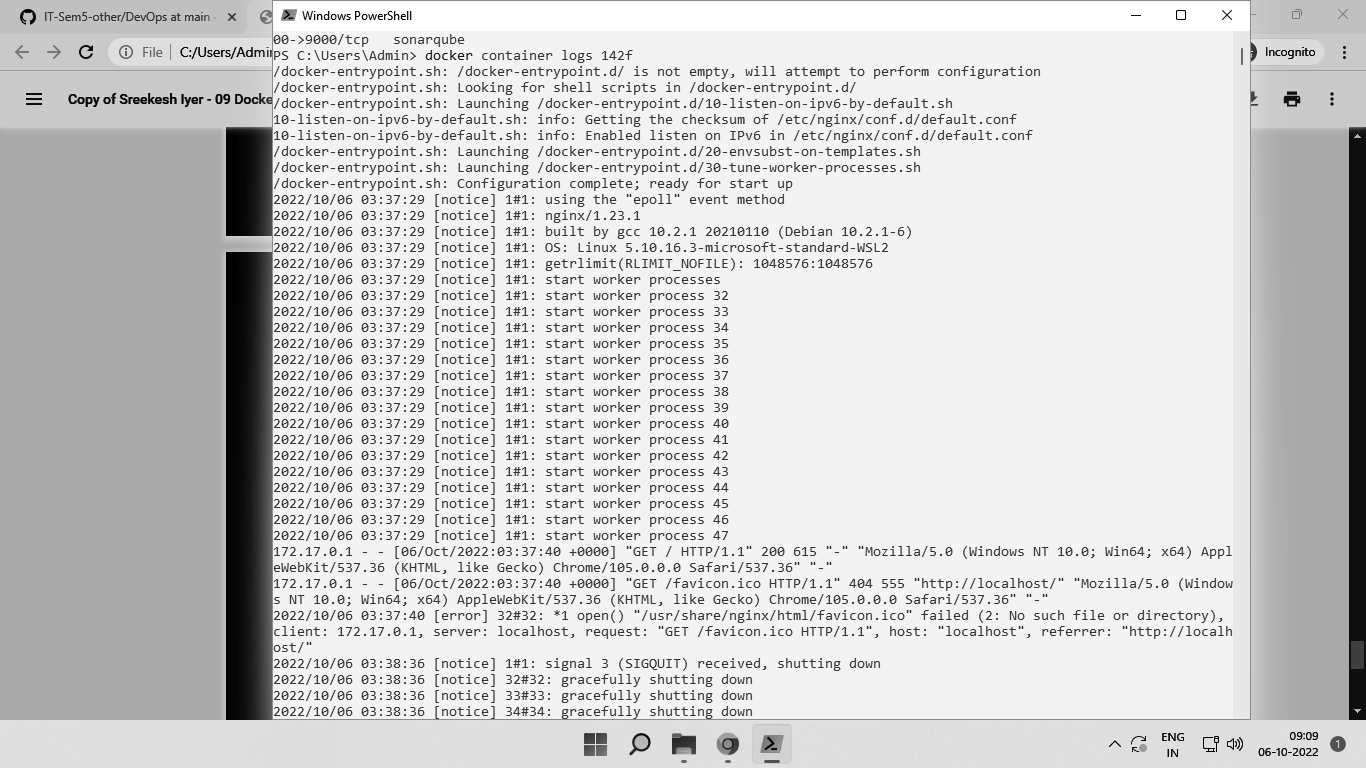
docker container ls -a



1. Show container logs

We can use the command logs to show logs for a specified container -

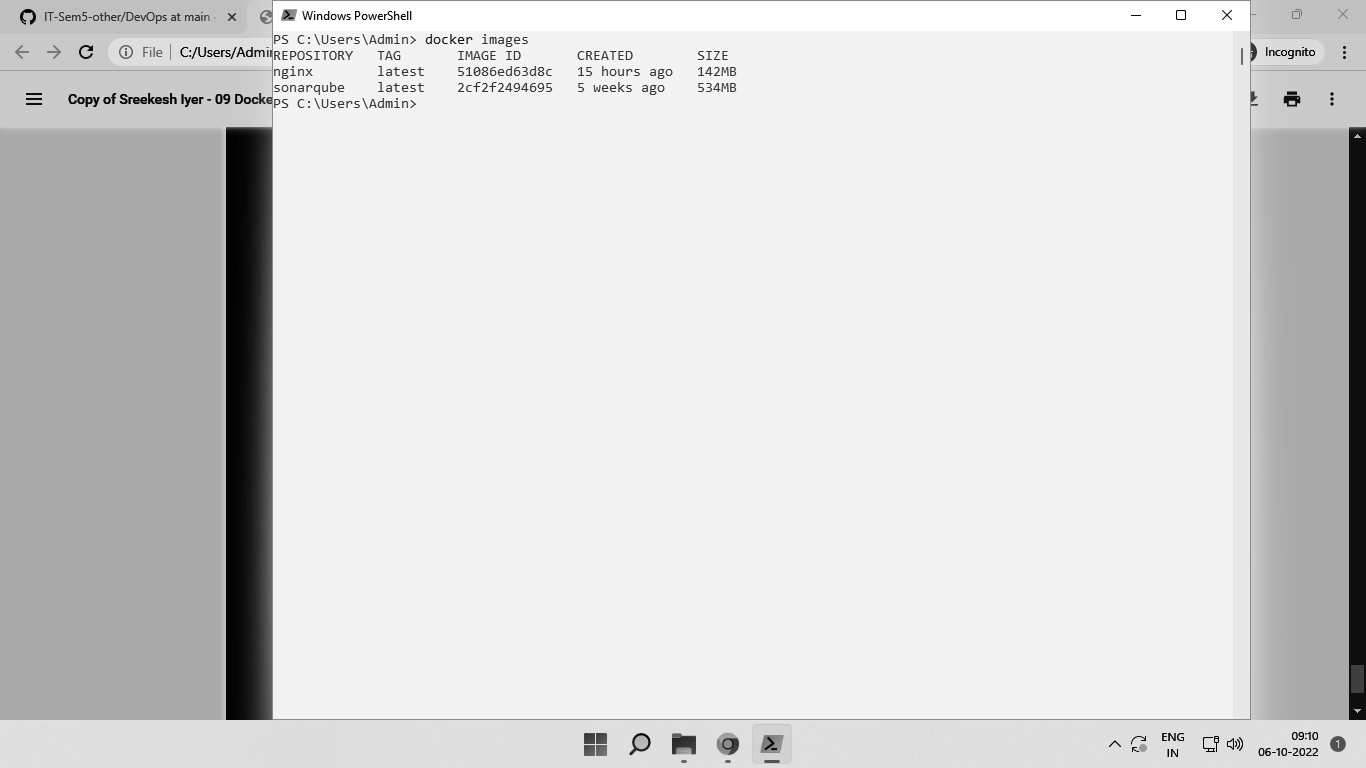
docker container logs <id>



1. Listing out Images

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

docker images



**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.