**Lab Manual- Container Based App Deployment using Docker**

Table of Contents

[1 OBJECTIVE 3](#_Toc75339189)

[2 PRE-REQUISISTE 3](#_Toc75339190)

[3 What is Docker and How it is different from Virtual Machine 3](#_Toc75339191)

[4 What is Docker Container ? 3](#_Toc75339192)

[5 Setup Up Docker 4](#_Toc75339193)

[5.1](#_Toc75339194) **[Create a Docker Account](#_Toc75339194)** [4](#_Toc75339194)

[6 Connect to Docker Playground 7](#_Toc75339195)

[7 Pull Images , Create Container , Install App in Container and Create the Image from Container and Push to Docker Repo 10](#_Toc75339196)

# OBJECTIVE

Deploying your software becomes a lot easier after Docker where you don’t have to worry about missing a system configuration or a prerequisite. In This Lab will cover the basics of running, starting, stopping, and removing Docker containers.

* Create an Account in Docker HUB
* Install the Docker Tool Box on windows
* Use Docker Playground for Labs
* Perform the Basic Management

# PRE-REQUISISTE

* Accounts in Azure
* A local Computer with 4 CPU, 16 GB RAM, 200 GB disk space

# What is Docker and How it is different from Virtual Machine

The main difference between them is that Docker is an **isolated process** that runs in your native OS while the virtual machine is a**complete isolated OS** that runs on top of your host OS which takes more time to load. So, Docker has benefits over virtual machines such as:

* Loading speed
* Small hardware resources required, unlike virtual machines.
* Running multiple Docker containers at the same time on the same OS.
* You can modify your container and deploy it or give the Docker file definition to a friend to start working on the same environment.

Actually, Docker is not a replacement for virtual machines, it comes to solve specific problems.

Suppose that your application needs 3 or more services which run on different operating systems so instead of running 3 virtual machines on the same host, you can run 3 containers smoothly on the same host. Sounds great!

# What is Docker Container ?

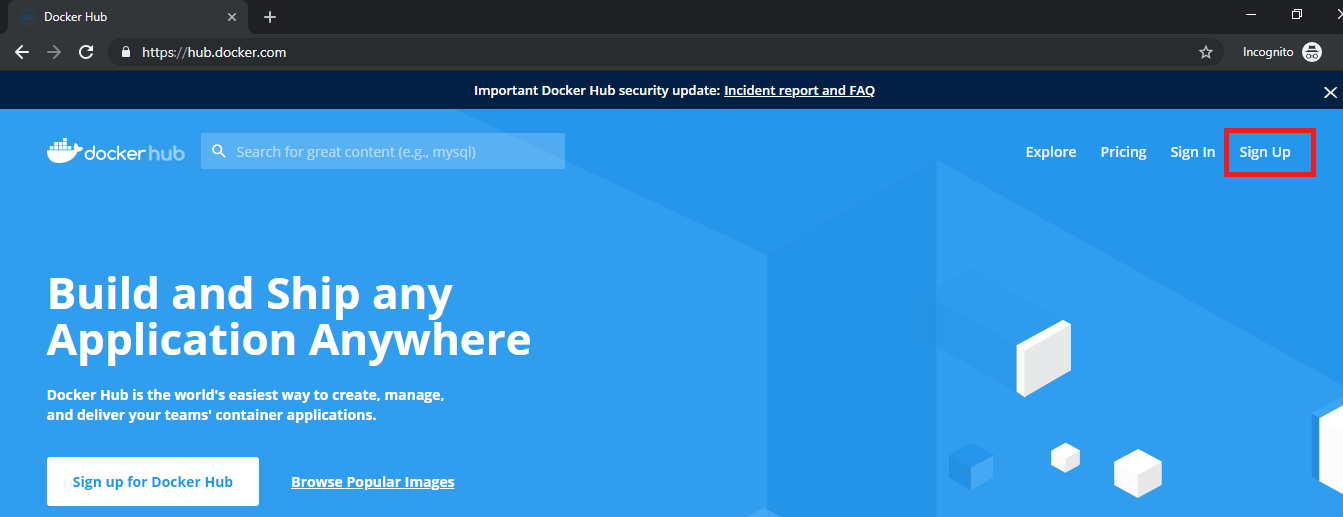
Containers offer a logical packaging mechanism in which applications can be abstracted from the environment in which they actually run. This decoupling allows container-based applications to be deployed easily and consistently, regardless of whether the target environment is a private data center, the public cloud, or even a developer’s personal laptop. This gives developers the ability to create predictable environments that are isolated from rest of the applications and can be run anywhere.

# Setup Up Docker

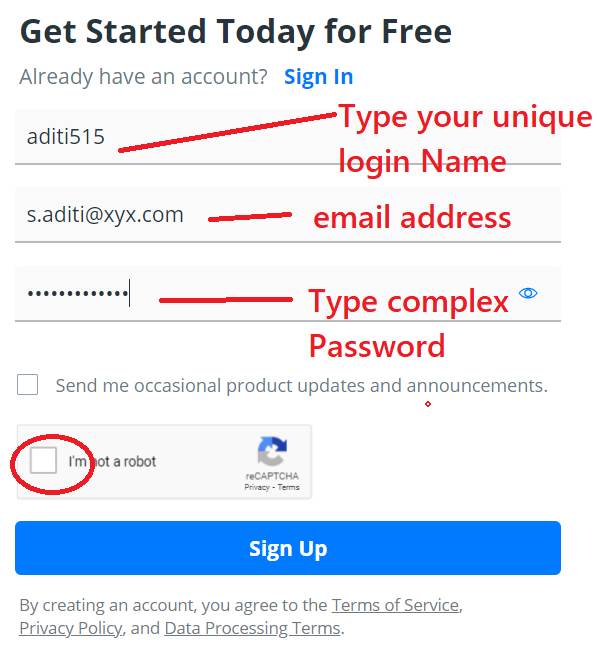
## **Create a Docker Account**

***Steps 1:*** Open the below URL to sign up the docker

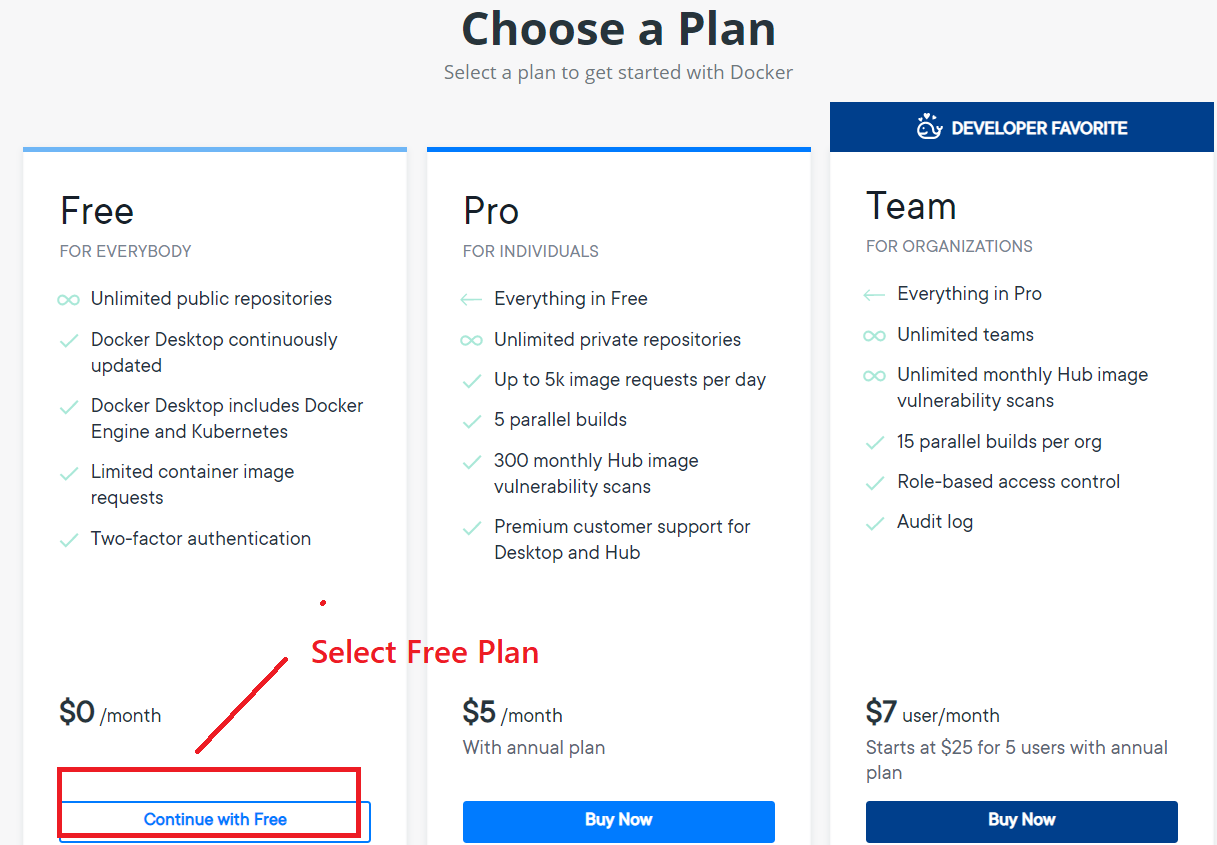
<https://hub.docker.com/>



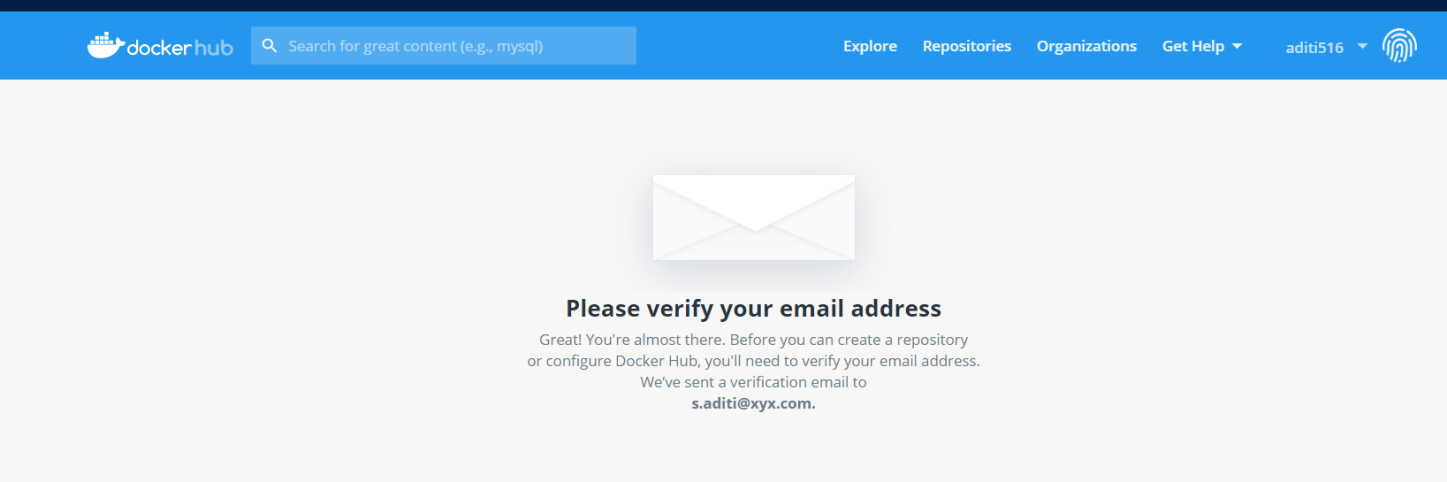
***Steps 2:*** Follow the Process of Signup as shown below



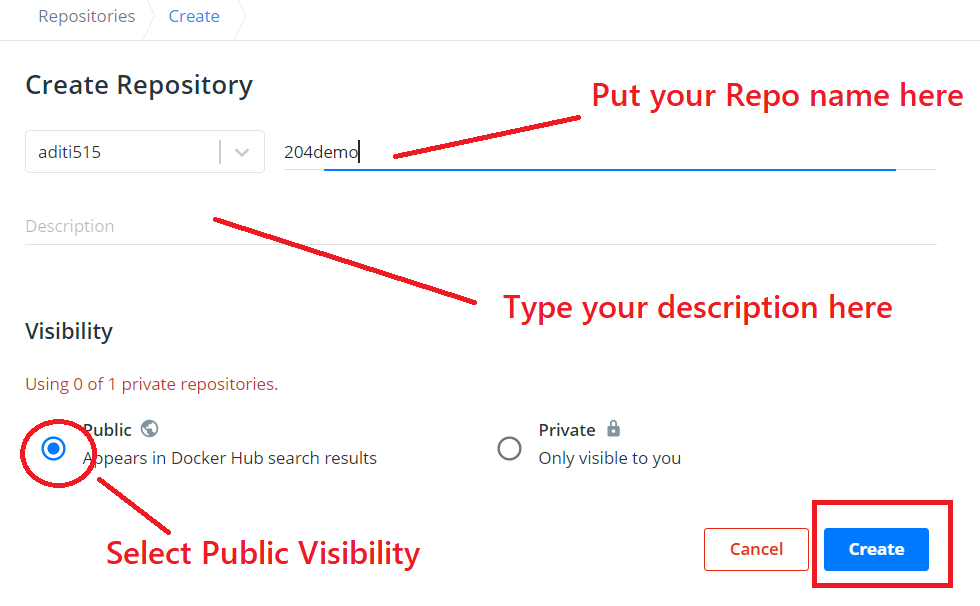
***Steps 3:*** Once you click signup you will present with screen similar to below Select free plan



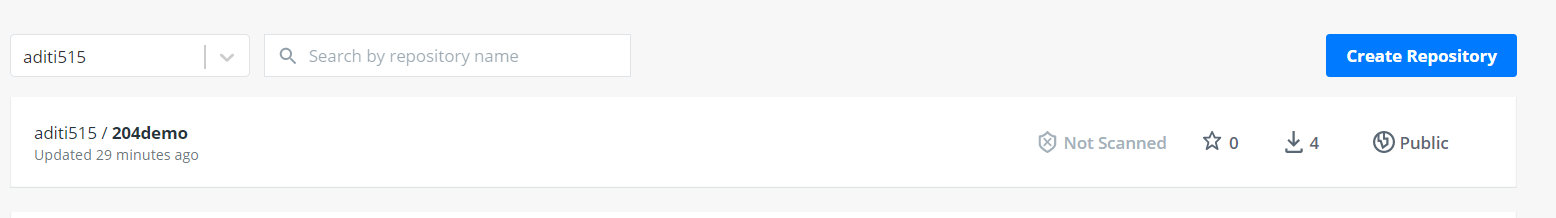
***Steps 4:*** Next Screen is to verify the email , go to your email and verify



***Steps 5:*** Once you verify the email, you will present the screen to create a Repository. Type **Repositories name** / **description** and **Scope** (Public /Private) and click **Create**



***Steps 7:*** Now your Repository should be available as shown in below

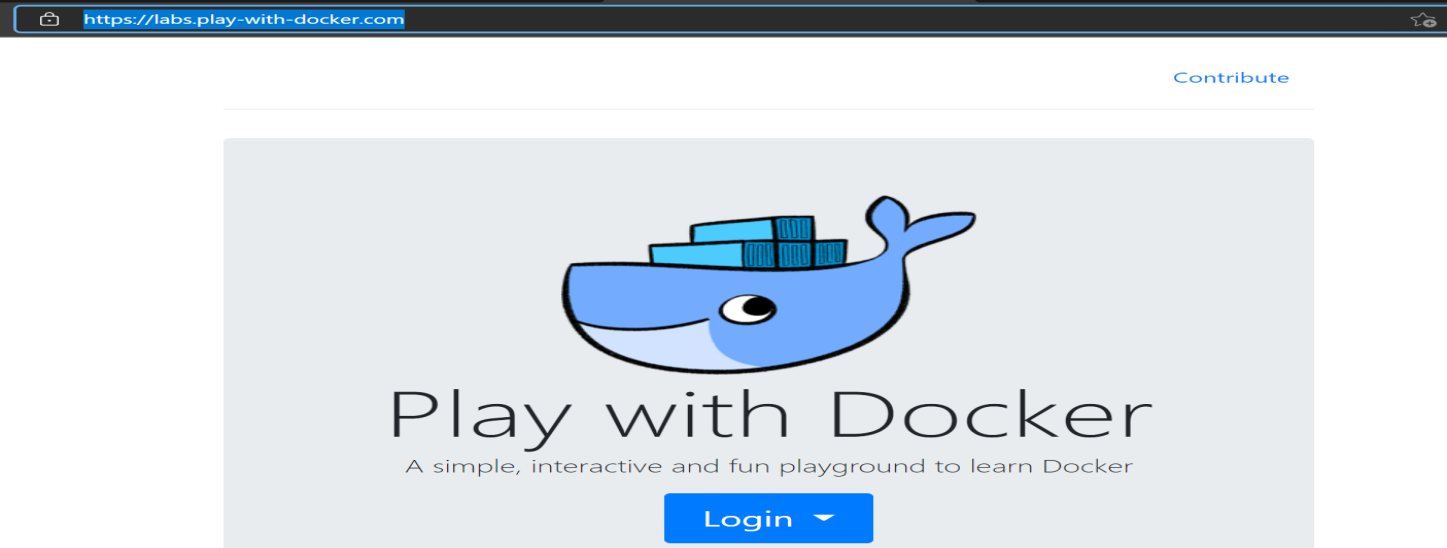


# Connect to Docker Playground

Playground provides a personalised sandboxed environment for you to learn and explore Docker.

***Steps 1:*** Open the Below URL in browser

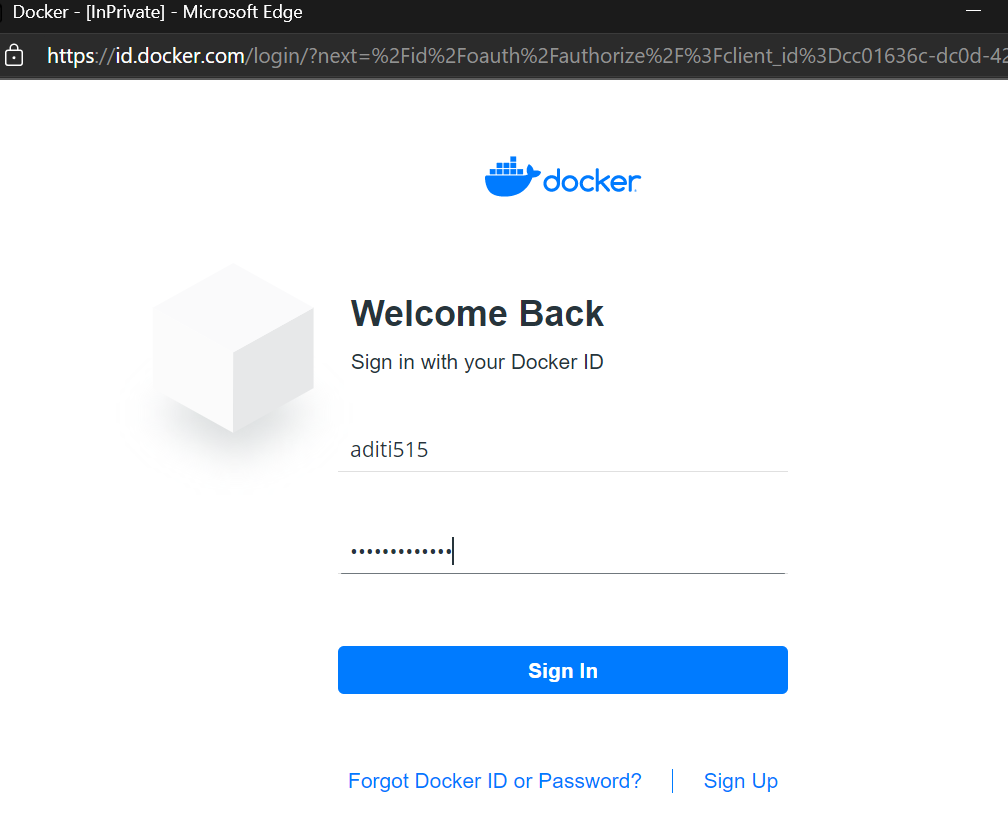
<https://labs.play-with-docker.com/>



***Steps 2:*** Click the Login Button and select Docker



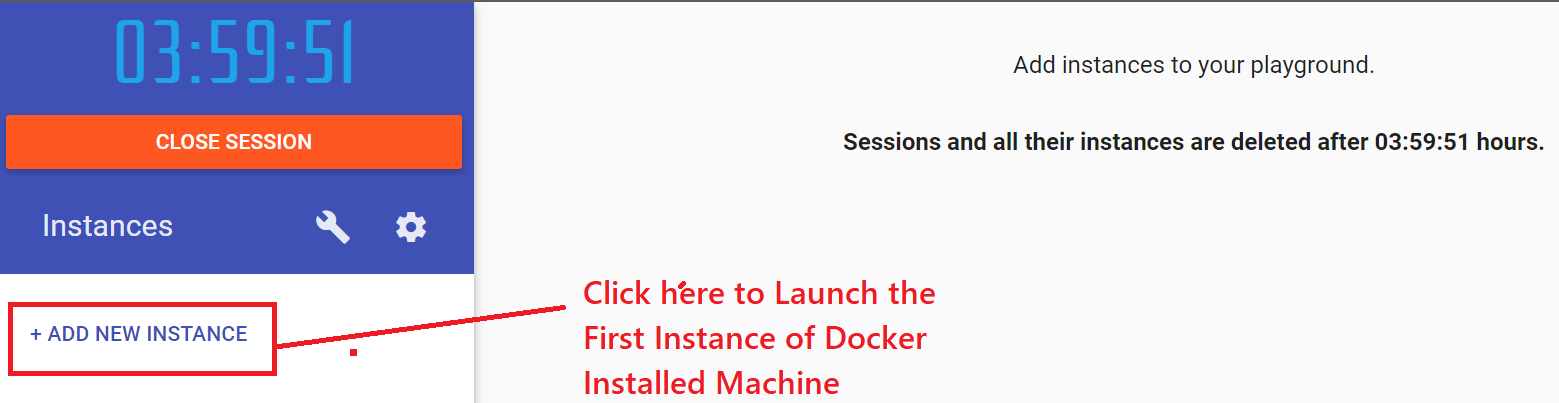
***Steps 3:*** Use your login and password you use to create Docker ID and click sign-in



***Steps 4:*** Now Click Start to Launch the Docker Playground



***Steps 5:*** On Left side of Panel there is Add New Instance



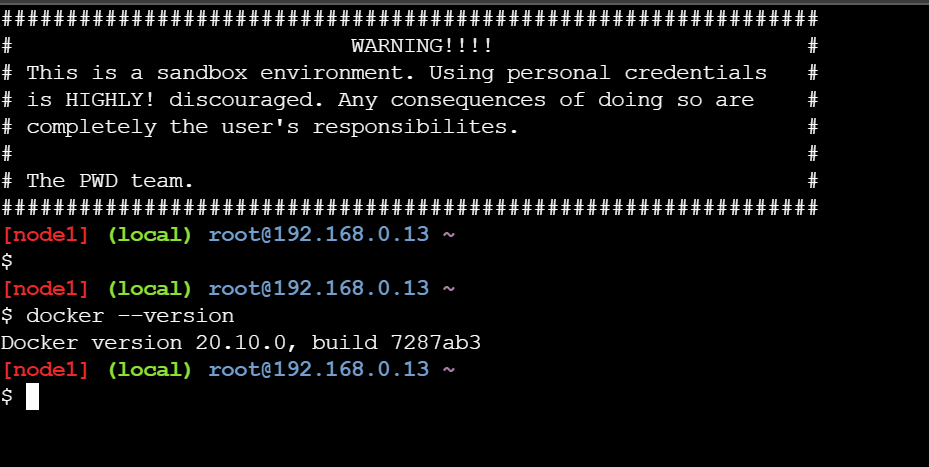
***Steps 6:*** Now new Black and white screen appear. Click inside and press **ALT + Enter.** It will launch the console in Full screen.



***Steps 7:*** Now Your console launch in full screen .If you want to increase the font Size just press CTRL and + .

***Steps 8:*** Now Type below command as below to check the version

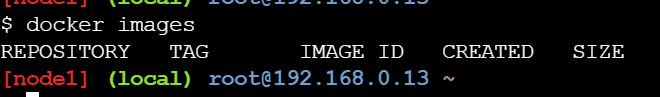
docker --version



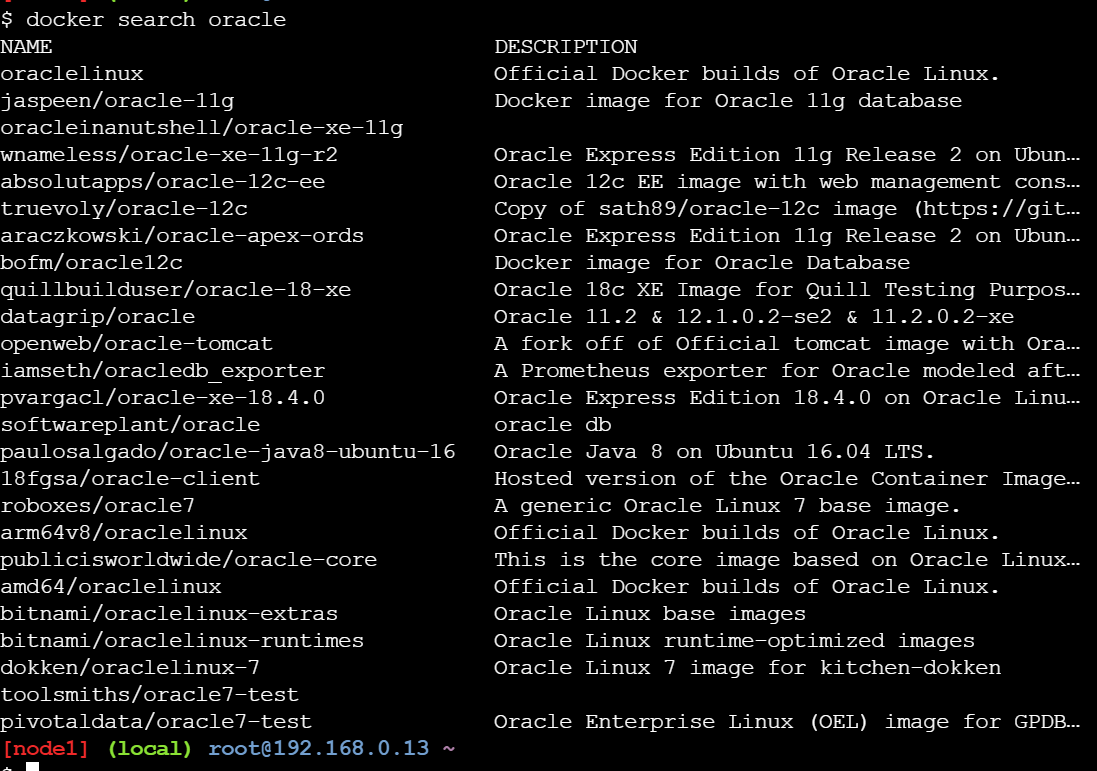
Now you are ready to play with all docker command

# Pull Images , Create Container , Install App in Container and Create the Image from Container and Push to Docker Repo

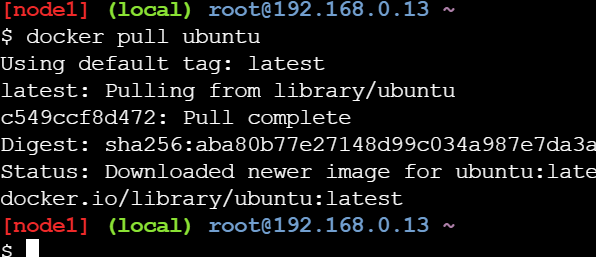
docker images



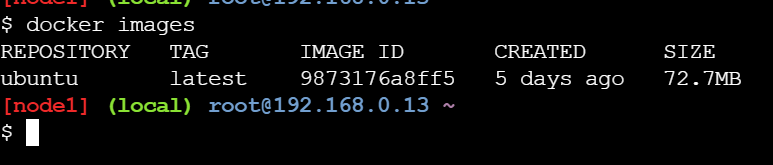
docker search oracle

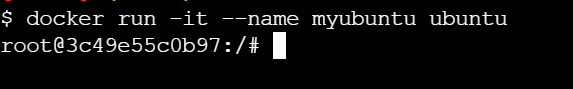


docker pull ubuntu

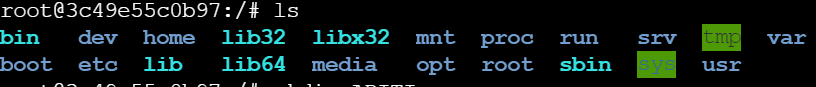
docker

docker images

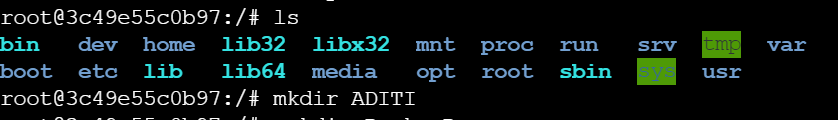




ls



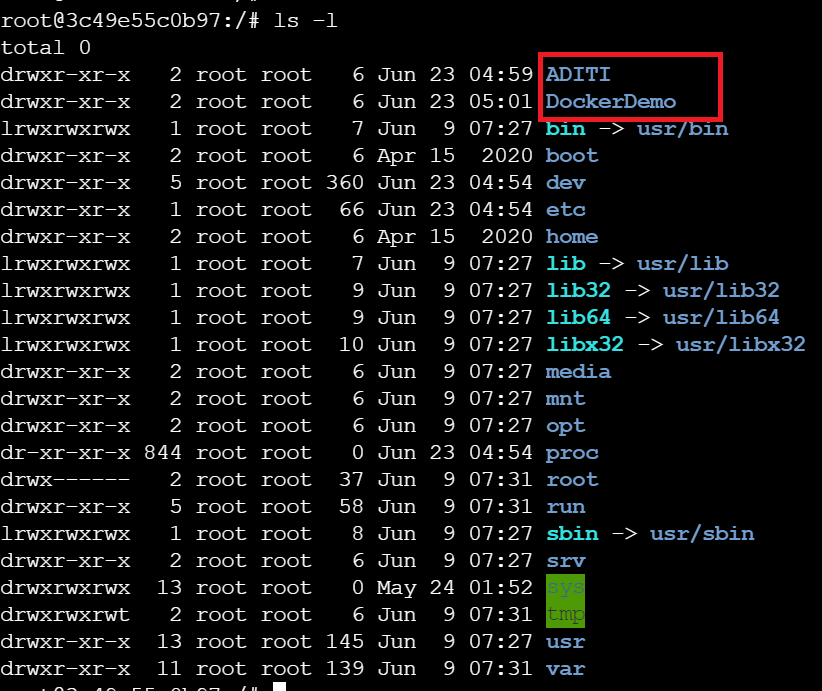
mkdir ADITI



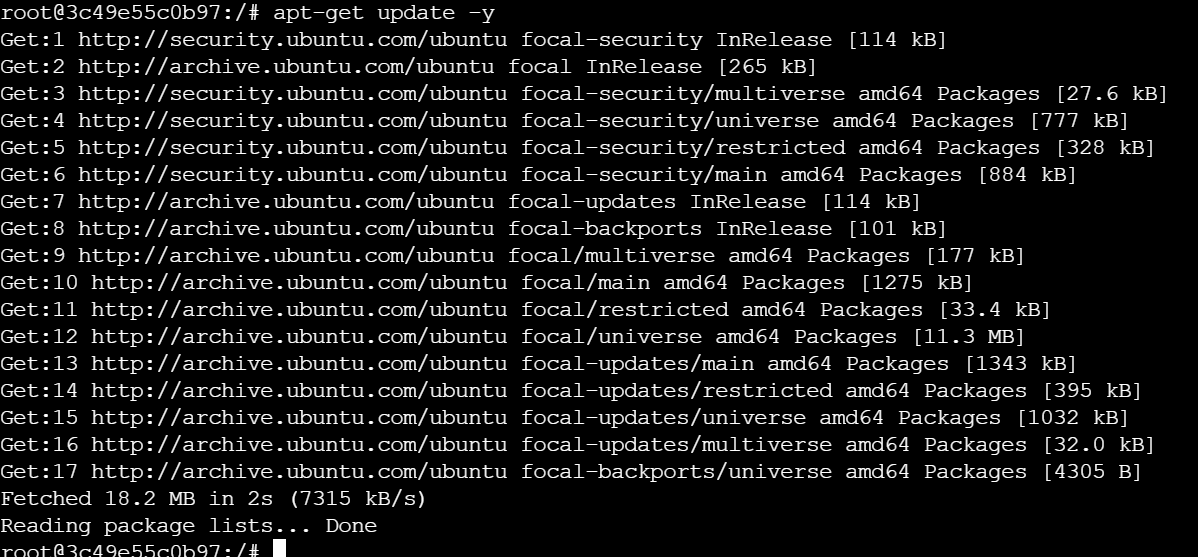
mkdir Docker-Demo



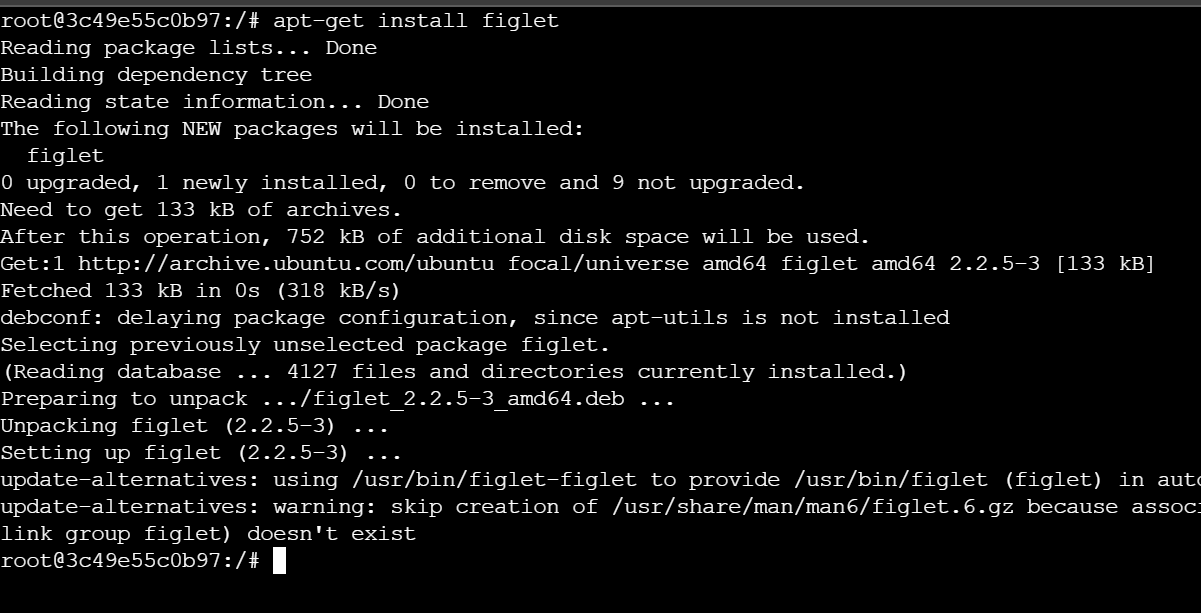
ls –l



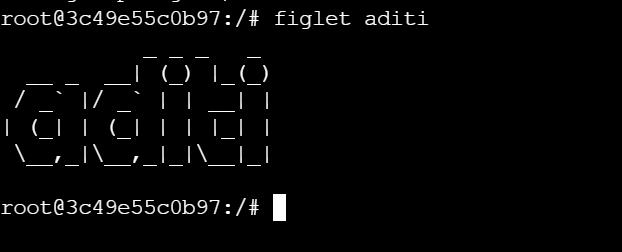
apt-get update



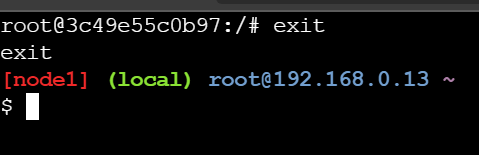
apt-get install figlet



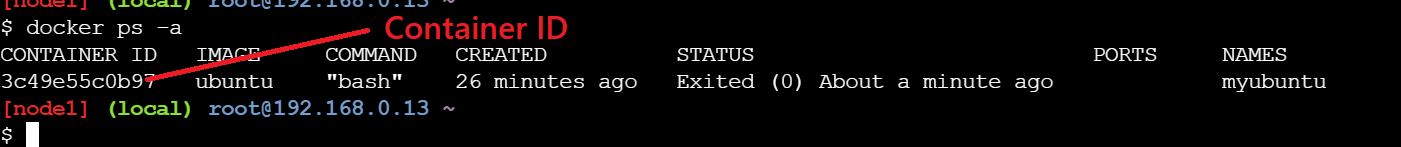
figlet Aditi



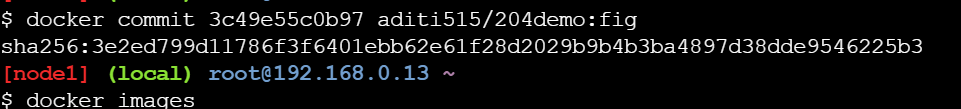
exit



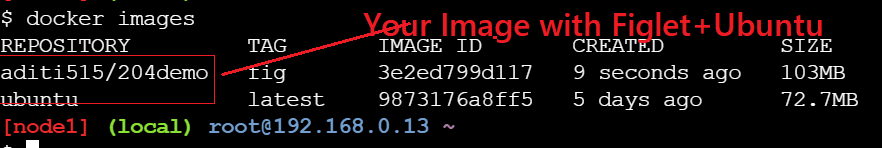
docker ps -a



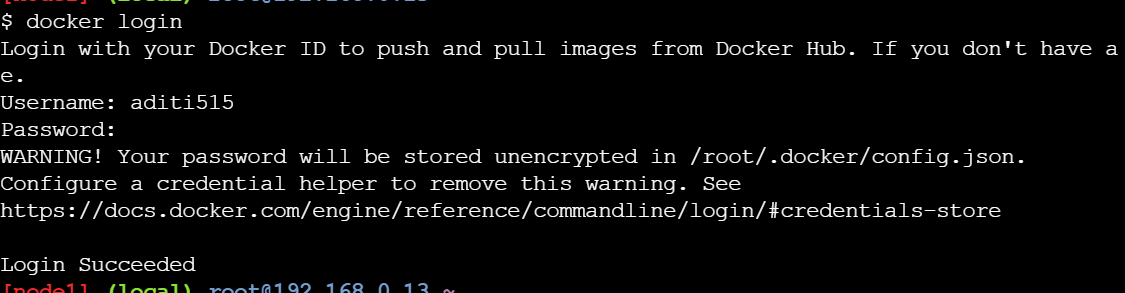
docker commit 3c49e55c0b97 aditi515/204demo:fig



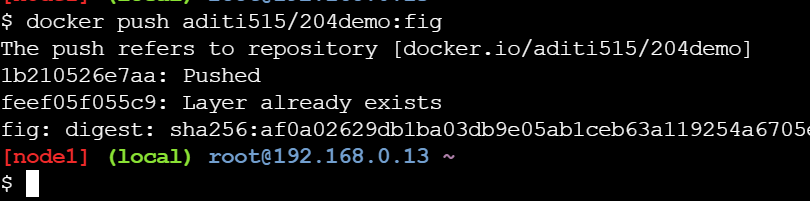
docker images



docker login



docker push aditi515/204demo:fig



Now you can go to Docker Hub and Check you will have your own customer Image which contain **Ubuntu OS + Figlet App**

