

Lab Manual- Container Based App Deployment using Docker

Prepared for:

Date:

Prepared by: Bipeen Sinha

Document Name: AZ400Container

Document Number AZ400361

Contributor:

Table of Contents

1	OBJECTIVE	3
2	PRE-REQUISISTE	3
3	What is Docker and How it is different from Virtual Machine	3
4	What is Docker Container ?	3
5	Setup Up Docker	4
5.1	Create a Docker Account	4
6	Connect to Docker Playground.....	7
7	Pull Images , Create Container , Install App in Container and Create the Image from Container and Push to Docker Repo	10

1 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

2 PRE-REQUISISTE

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

3 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!

4 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,

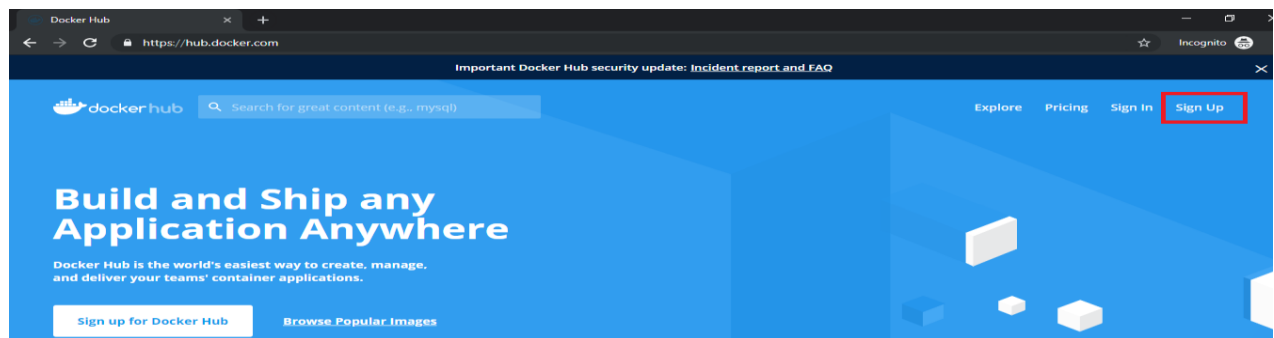
the public cloud, or even a developer's personal laptop. This gives developers the ability to create predictable environments that are isolated from rest of the applications and can be run anywhere.

5 Setup Up Docker

5.1 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

Get Started Today for Free

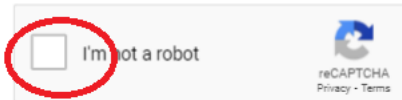
Already have an account? [Sign In](#)

aditi515  Type your unique login Name

s.aditi@xyx.com  email address

.....|  Type complex Password 

☐ Send me occasional product updates and announcements.



Sign Up

By creating an account, you agree to the [Terms of Service](#), [Privacy Policy](#), and [Data Processing Terms](#).

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

Choose a Plan

Select a plan to get started with Docker

Free

FOR EVERYBODY

- ∞ Unlimited public repositories
- ✓ Docker Desktop continuously updated
- ✓ Docker Desktop includes Docker Engine and Kubernetes
- ✓ Limited container image requests
- ✓ Two-factor authentication

\$0 /month

Continue with Free

Pro

FOR INDIVIDUALS

← Everything in Free

- ∞ Unlimited private repositories
- ✓ Up to 5k image requests per day
- ✓ 5 parallel builds
- ✓ 300 monthly Hub image vulnerability scans
- ✓ Premium customer support for Desktop and Hub

\$5 /month

With annual plan

Buy Now

Team

FOR ORGANIZATIONS

← Everything in Pro

- ∞ Unlimited teams
- ∞ Unlimited monthly Hub image vulnerability scans
- ✓ 15 parallel builds per org
- ✓ Role-based access control
- ✓ Audit log

\$7 user/month


Starts at \$25 for 5 users with annual plan

Buy Now


DEVELOPER FAVORITE

Select Free Plan

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



[Explore](#) [Repositories](#) [Organizations](#) [Get Help](#) [aditi516](#)



Please verify your email address

Great! You're almost there. Before you can create a repository or configure Docker Hub, you'll need to verify your email address. We've sent a verification email to s.aditi@xyx.com.

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**

Repositories > Create


Create Repository


aditi515 | 204demo **Put your Repo name here**

Description **Type your description here**

Visibility

Using 0 of 1 private repositories.

☒ **Public**  Appears in Docker Hub search results

☐ **Private**  Only visible to you





Select Public Visibility

Cancel Create

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

aditi515 | Search by repository name Create Repository

aditi515 / 204demo
Updated 29 minutes ago

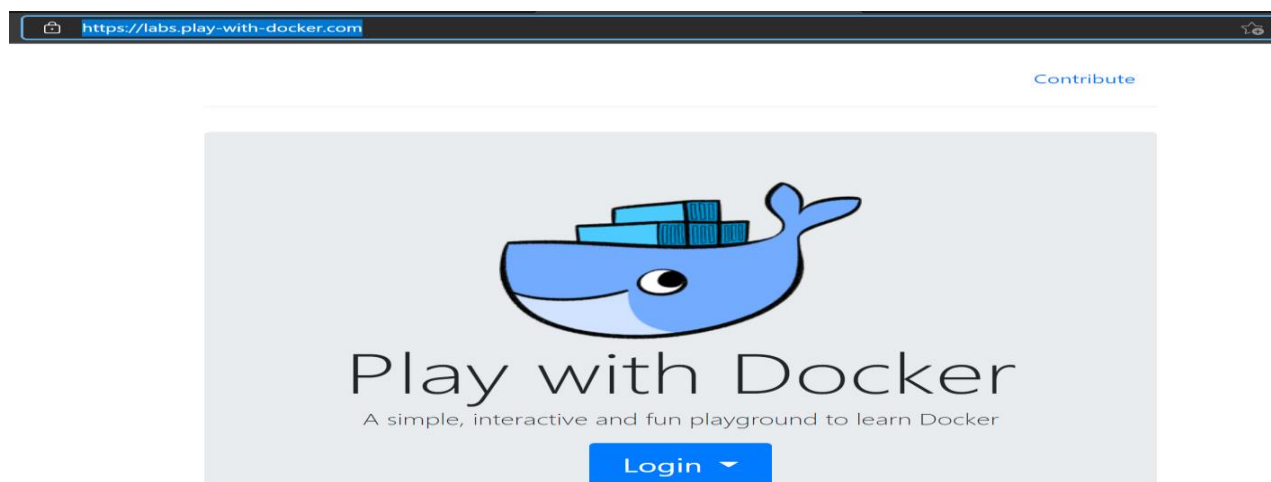
 Not Scanned  0  4  Public

6 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

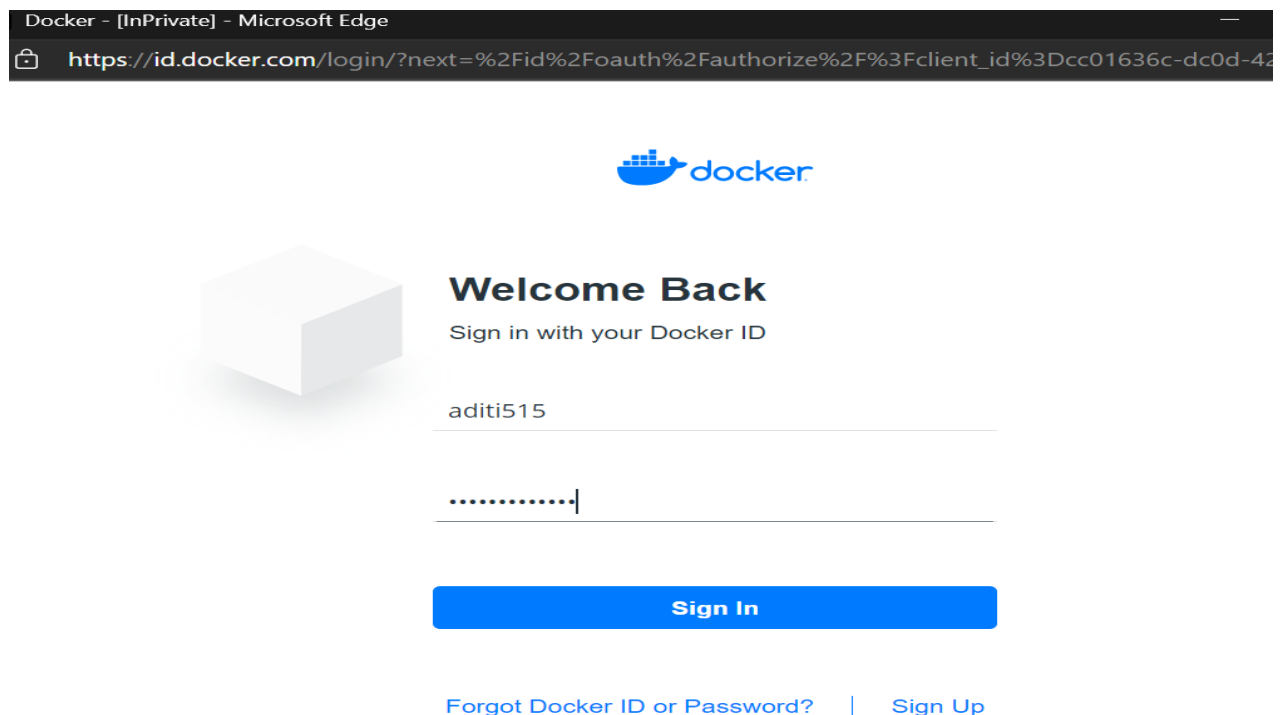
<https://labs.play-with-docker.com/>



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

A screenshot of the Play with Docker interface. On the left is a dark blue sidebar with a timer at the top showing '03:59:51'. Below the timer is an orange 'CLOSE SESSION' button. Underneath is the word 'Instances' with a wrench and gear icon. At the bottom of the sidebar is a button labeled '+ ADD NEW INSTANCE', which is highlighted with a red rectangular box. A red arrow points from this box to the right. On the right side of the interface, there is a light gray area with the text 'Add instances to your playground.' and 'Sessions and all their instances are deleted after 03:59:51 hours.' Below this, the text 'Click here to Launch the First Instance of Docker Installed Machine' is written in red.

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

A screenshot of the Play with Docker interface showing a Linux-based console window. The top part of the interface is the same as in the previous screenshot, but the timer now shows '03:56:56'. Below the sidebar, there is a list of instances with one instance named 'node1' at IP '192.168.0.13'. To the right, a detailed view of the selected instance is shown. It includes the instance ID 'c39bi5vq_c39bji7njsv000fab440', the IP '192.168.0.13', and buttons for 'OPEN PORT', 'DELETE', and 'EDITOR'. Below this, the SSH command is shown: 'ssh ip172-18-0-36-c39bi5vqf8u0009dgle0@direct.labs.play-'. A red arrow points from the text 'Linux Based Console' to the console window. The console window itself is black with white text, displaying a warning message: 'WARNING!!!! This is a sandbox environment. Using personal credentials is HIGHLY! discouraged. Any consequences of doing so are completely the user's responsibilities. The PWD team.' followed by a prompt '[node1] (local) root@192.168.0.13 ~' and a dollar sign '\$'. A red arrow points from the text 'Click Here and Press Alt+Enter' to the console window.

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

```
#####  
#                               WARNING!!!!                               #  
# This is a sandbox environment. Using personal credentials             #  
# is HIGHLY! discouraged. Any consequences of doing so are             #  
# completely the user's responsibilites.                                #  
#                                                                       #  
# The PWD team.                                                         #  
#####  
[node1] (local) root@192.168.0.13 ~  
$  
[node1] (local) root@192.168.0.13 ~  
$ docker --version  
Docker version 20.10.0, build 7287ab3  
[node1] (local) root@192.168.0.13 ~  
$
```

Now you are ready to play with all docker command

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

docker images

```
[node1] (local) root@192.168.0.13 ~  
$ docker images  
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE  
[node1] (local) root@192.168.0.13 ~
```

docker search oracle

```
$ docker search oracle
```

NAME	DESCRIPTION
oraclelinux	Official Docker builds of Oracle Linux.
jaspeen/oracle-11g	Docker image for Oracle 11g database
oracleinanutshell/oracle-xe-11g	
wnameless/oracle-xe-11g-r2	Oracle Express Edition 11g Release 2 on Ubun...
absolutapps/oracle-12c-ee	Oracle 12c EE image with web management cons...
truevoly/oracle-12c	Copy of sath89/oracle-12c image (https://git...
araczkowski/oracle-apex-ords	Oracle Express Edition 11g Release 2 on Ubun...
bofm/oracle12c	Docker image for Oracle Database
quillbuilduser/oracle-18-xe	Oracle 18c XE Image for Quill Testing Purpos...
datagrip/oracle	Oracle 11.2 & 12.1.0.2-se2 & 11.2.0.2-xe
openweb/oracle-tomcat	A fork off of Official tomcat image with Ora...
iamseth/oracledb_exporter	A Prometheus exporter for Oracle modeled aft...
pvargacl/oracle-xe-18.4.0	Oracle Express Edition 18.4.0 on Oracle Linu...
softwareplant/oracle	oracle db
paulosalgado/oracle-java8-ubuntu-16	Oracle Java 8 on Ubuntu 16.04 LTS.
l8fgsa/oracle-client	Hosted version of the Oracle Container Image...
robboxes/oracle7	A generic Oracle Linux 7 base image.
arm64v8/oraclelinux	Official Docker builds of Oracle Linux.
publicisworldwide/oracle-core	This is the core image based on Oracle Linux...
amd64/oraclelinux	Official Docker builds of Oracle Linux.
bitnami/oraclelinux-extras	Oracle Linux base images
bitnami/oraclelinux-runtimes	Oracle Linux runtime-optimized images
dokken/oraclelinux-7	Oracle Linux 7 image for kitchen-dokken
toolsmiths/oracle7-test	
pivotaldata/oracle7-test	Oracle Enterprise Linux (OEL) image for GPDB...

```
[node1] (local) root@192.168.0.13 ~
```

docker pull ubuntu

```
[node1] (local) root@192.168.0.13 ~
$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
c549ccf8d472: Pull complete
Digest: sha256:aba80b77e27148d99c034a987e7da3a
Status: Downloaded newer image for ubuntu:late
docker.io/library/ubuntu:latest
[node1] (local) root@192.168.0.13 ~
$
```

docker images

```
[node1] (local) root@192.168.0.13 ~
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	9873176a8ff5	5 days ago	72.7MB

```
[node1] (local) root@192.168.0.13 ~
$
```

docker run -it --name myubuntu ubuntu

```
$ docker run -it --name myubuntu ubuntu
root@3c49e55c0b97:/#
```

ls

```
root@3c49e55c0b97:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
```

mkdir ADITI

```
root@3c49e55c0b97:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@3c49e55c0b97:/# mkdir ADITI
```

mkdir Docker-Demo

```
root@3c49e55c0b97:/# mkdir DockerDemo
```

ls -l

```
root@3c49e55c0b97:/# ls -l
total 0
drwxr-xr-x  2 root root    6 Jun 23 04:59 ADITI
drwxr-xr-x  2 root root    6 Jun 23 05:01 DockerDemo
lrwxrwxrwx  1 root root    7 Jun  9 07:27 bin -> usr/bin
drwxr-xr-x  2 root root   66 Apr 15 2020 boot
drwxr-xr-x  5 root root  360 Jun 23 04:54 dev
drwxr-xr-x  1 root root   66 Jun 23 04:54 etc
drwxr-xr-x  2 root root   66 Apr 15 2020 home
lrwxrwxrwx  1 root root    7 Jun  9 07:27 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Jun  9 07:27 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Jun  9 07:27 lib64 -> usr/lib64
lrwxrwxrwx  1 root root   10 Jun  9 07:27 libx32 -> usr/libx32
drwxr-xr-x  2 root root    6 Jun  9 07:27 media
drwxr-xr-x  2 root root    6 Jun  9 07:27 mnt
drwxr-xr-x  2 root root    6 Jun  9 07:27 opt
dr-xr-xr-x 844 root root    0 Jun 23 04:54 proc
drwx----- 2 root root   37 Jun  9 07:31 root
drwxr-xr-x  5 root root   58 Jun  9 07:31 run
lrwxrwxrwx  1 root root    8 Jun  9 07:27 sbin -> usr/sbin
drwxr-xr-x  2 root root    6 Jun  9 07:27 srv
drwxrwxrwx 13 root root    0 May 24 01:52 sys
drwxrwxrwt  2 root root    6 Jun  9 07:31 tmp
drwxr-xr-x 13 root root  145 Jun  9 07:27 usr
drwxr-xr-x 11 root root  139 Jun  9 07:31 var
```

apt-get update -y

```

root@3c49e55c0b97:/# apt-get update -y
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [27.6 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [777 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [328 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [884 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [177 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/main amd64 Packages [1275 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [33.4 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal/universe amd64 Packages [11.3 MB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1343 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [395 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1032 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [32.0 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [4305 B]
Fetched 18.2 MB in 2s (7315 kB/s)
Reading package lists... Done
root@3c49e55c0b97:/#

```

apt-get install figlet

```

root@3c49e55c0b97:/# apt-get install figlet
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  figlet
0 upgraded, 1 newly installed, 0 to remove and 9 not upgraded.
Need to get 133 kB of archives.
After this operation, 752 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 figlet amd64 2.2.5-3 [133 kB]
Fetched 133 kB in 0s (318 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package figlet.
(Reading database ... 4127 files and directories currently installed.)
Preparing to unpack .../figlet_2.2.5-3_amd64.deb ...
Unpacking figlet (2.2.5-3) ...
Setting up figlet (2.2.5-3) ...
update-alternatives: using /usr/bin/figlet-figlet to provide /usr/bin/figlet (figlet) in auto
update-alternatives: warning: skip creation of /usr/share/man/man6/figlet.6.gz because associ
link group figlet) doesn't exist
root@3c49e55c0b97:/#

```

figlet Aditi

```

root@3c49e55c0b97:/# figlet aditi
  _ _ _ _ _
 / _ \ _ \ / _ \ _ \
| ( _ | | ( _ | | |
 \ _ / _ \ \ _ / _ \
root@3c49e55c0b97:/#

```

exit

```
root@3c49e55c0b97:/# exit
exit
[node1] (local) root@192.168.0.13 ~
$
```

docker ps -a

```
[node1] (local) root@192.168.0.13 ~
$ docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS          NAMES
3c49e55c0b97   ubuntu    "bash"                  26 minutes ago   Exited (0)      About a minute ago   myubuntu
[node1] (local) root@192.168.0.13 ~
$
```

docker commit 3c49e55c0b97 aditi515/204demo:fig

```
$ docker commit 3c49e55c0b97 aditi515/204demo:fig
sha256:3e2ed799d11786f3f6401ebb62e61f28d2029b9b4b3ba4897d38dde9546225b3
[node1] (local) root@192.168.0.13 ~
$ docker images
```

docker images

```
$ docker images
REPOSITORY      TAG       IMAGE ID       CREATED        SIZE
aditi515/204demo fig       3e2ed799d117   9 seconds ago  103MB
ubuntu          latest    9873176a8ff5   5 days ago    72.7MB
[node1] (local) root@192.168.0.13 ~
```

docker login

```
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a
e.
Username: aditi515
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[node1] (local) root@192.168.0.13 ~
```


docker push aditi515/204demo:fig


```
$ docker push aditi515/204demo:fig
The push refers to repository [docker.io/aditi515/204demo]
1b210526e7aa: Pushed
feef05f055c9: Layer already exists
fig: digest: sha256:af0a02629db1ba03db9e05ab1ceb63a119254a6705e
[node1] (local) root@192.168.0.13 ~
$
```


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

Repositories > aditi515 / 204demo > **Your Docker Repo** Using 0 of 1 private repositories.

General Tags Builds

 /

This repository does not have a description 

 Last pushed: never


Docker commands [Public View](#)

To push a new tag to this repository,

```
docker push aditi515/204demo:tagname
```

Tag Activity

This repository contains 2 tag(s).

TAG	OS	PUSHED
fig		2 minutes ago

Your Image now in Repo

Any other user can use this command in their docker client terminal to pull your image.