THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXPERIMENT NO. 9

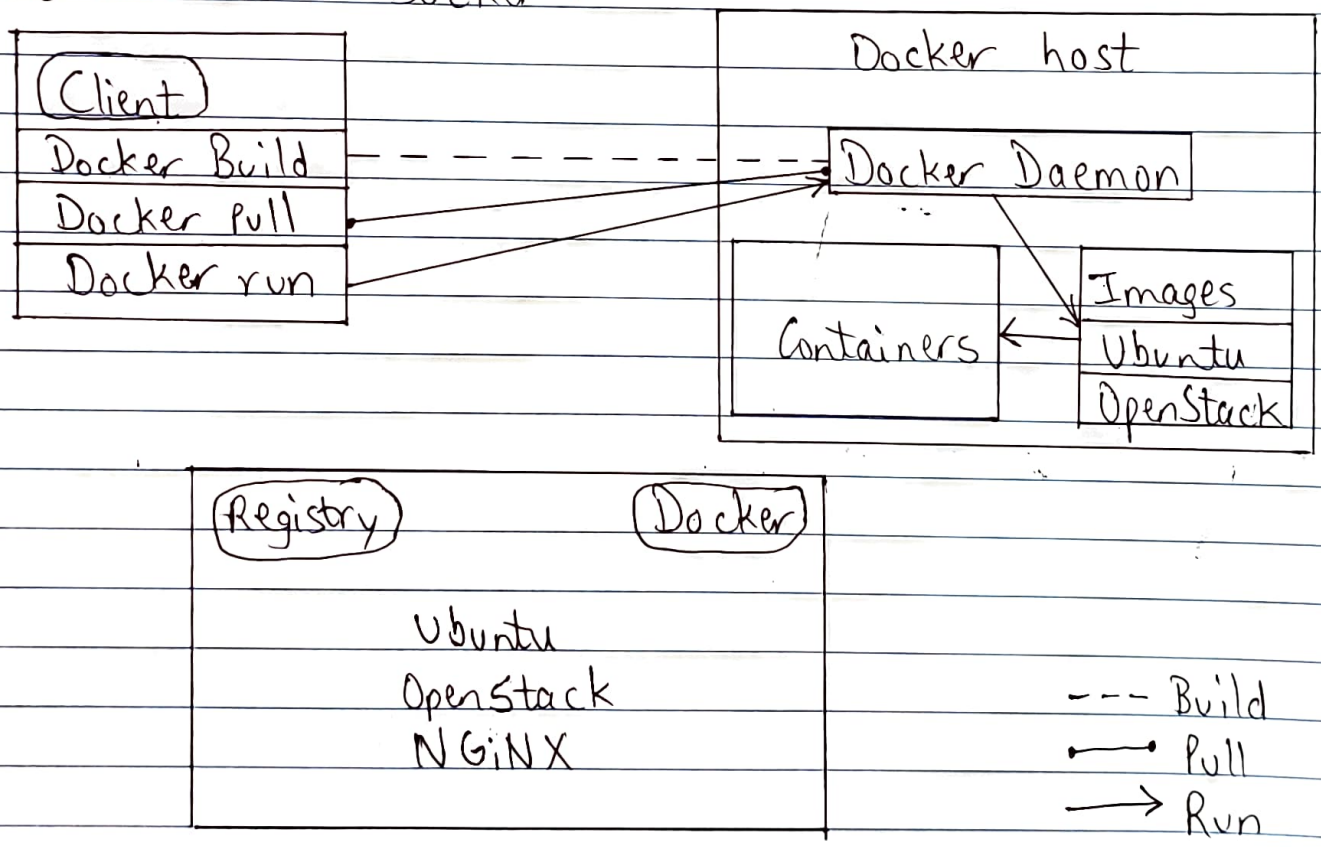
Aim :- To study and implement containerization using Docker.

Theory :-

Docker :

Docker is a platform that enables developers to automate the deployment and management of applications, within lightweight, portable containers. These containers ensure consistency across different environments, allowing applications to run seamlessly on various systems.

Architecture of Docker :



THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

Benefits of Containerization :-

i) Portability
ii) Efficiency
iii) Agility
iv) Faster Delivery
v) Improved Security
vi) Flexibility

Container	Images
A container is a running instance of a Docker image.	An image is a static, immutable snapshot or template used to create containers.
Containers have state and can be in various states like running, paused, etc.	Once created, an image does not change its state. They are stateless.
Created on the local system during runtime.	Stored in registry for distribution and sharing.

Conclusion :-

Thus, I was able to understand and implement containerization using Docker.

20/3/24

Rishab Mandal
2103110
C23

Cloud Computing Experiment 9

Aim: To study and Implement Containerization using Docker.

Theory:

Introduction:

In recent years, the landscape of software development and deployment has undergone a significant transformation, driven by the need for more efficient, scalable, and portable solutions. Traditional approaches to application deployment often face challenges related to dependency management, environment consistency, and scalability. In response to these challenges, containerization has emerged as a powerful paradigm, offering a lightweight, portable, and efficient solution for packaging, distributing, and running applications.

At the forefront of containerization technology is Docker, a platform that has revolutionized the way developers build, ship, and run applications. Docker provides a standardized environment for packaging applications and their dependencies into isolated containers, which can then be deployed across various environments seamlessly. With Docker, developers can encapsulate their applications along with all required dependencies, ensuring consistency across different development, testing, and production environments.

This experiment aims to explore and implement containerization using Docker, a widely adopted tool in the field of DevOps and software development. By understanding Docker's architecture, benefits of containerization, and key concepts such as containers, images, and Docker files, participants will gain

insights into modern software deployment practices. Through practical implementation and comparison with traditional approaches such as virtual machines, participants will appreciate the efficiency, scalability, and portability offered by Docker containers.

In this write-up, we will delve into the fundamentals of Docker, explore its architecture, discuss the advantages of containerization, and elucidate key Docker concepts. Additionally, we will provide a step-by-step guide to implementing containerization using Docker, accompanied by screenshots to illustrate the process. By the end of this experiment, participants will have a solid understanding of Docker and its role in modern software development workflows.

1) Docker architecture:

Docker is a platform that allows you to develop, deploy, and run applications in containers. It uses a client-server architecture. The Docker client communicates with the Docker daemon, which does the heavy lifting of building, running, and distributing containers. The Docker daemon manages Docker objects such as images, containers, networks, and volumes.

2) Benefits of Containerization:

- ****Portability:**** Containers encapsulate everything an application needs to run, making it easy to move between environments.

- ****Isolation:**** Containers provide isolation for applications, preventing conflicts between dependencies.

- ****Efficiency:**** Containers share the host OS kernel, reducing overhead and resource usage compared to virtual machines.

- ****Scalability:**** Containers can be quickly scaled up or down to meet demand, enabling efficient resource utilization.

3) Explain following w.r.t Docker:

- **Container:** A container is a lightweight, standalone, executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and dependencies.

- **Images:** An image is a read-only template used to create containers. It contains the application code along with all its dependencies.

- **Dockerfile:** A Dockerfile is a text document that contains instructions for building a Docker image. It specifies the base image, environment variables, dependencies, and commands to run when the container starts.

4) Comparison:

- **Container vs. Virtual Machine:** Containers share the host OS kernel, while virtual machines have their own OS kernel. This makes containers lighter and more efficient than virtual machines.

- **Images vs. Containers:** An image is a template for creating containers, while a container is a running instance of an image. Multiple containers can be created from the same image.

Steps for Implementing Containerization using Docker:

1. Install Docker on your system.

2. Write a Docker file specifying the application's dependencies and commands.

3. Build the Docker image using the `docker build` command.

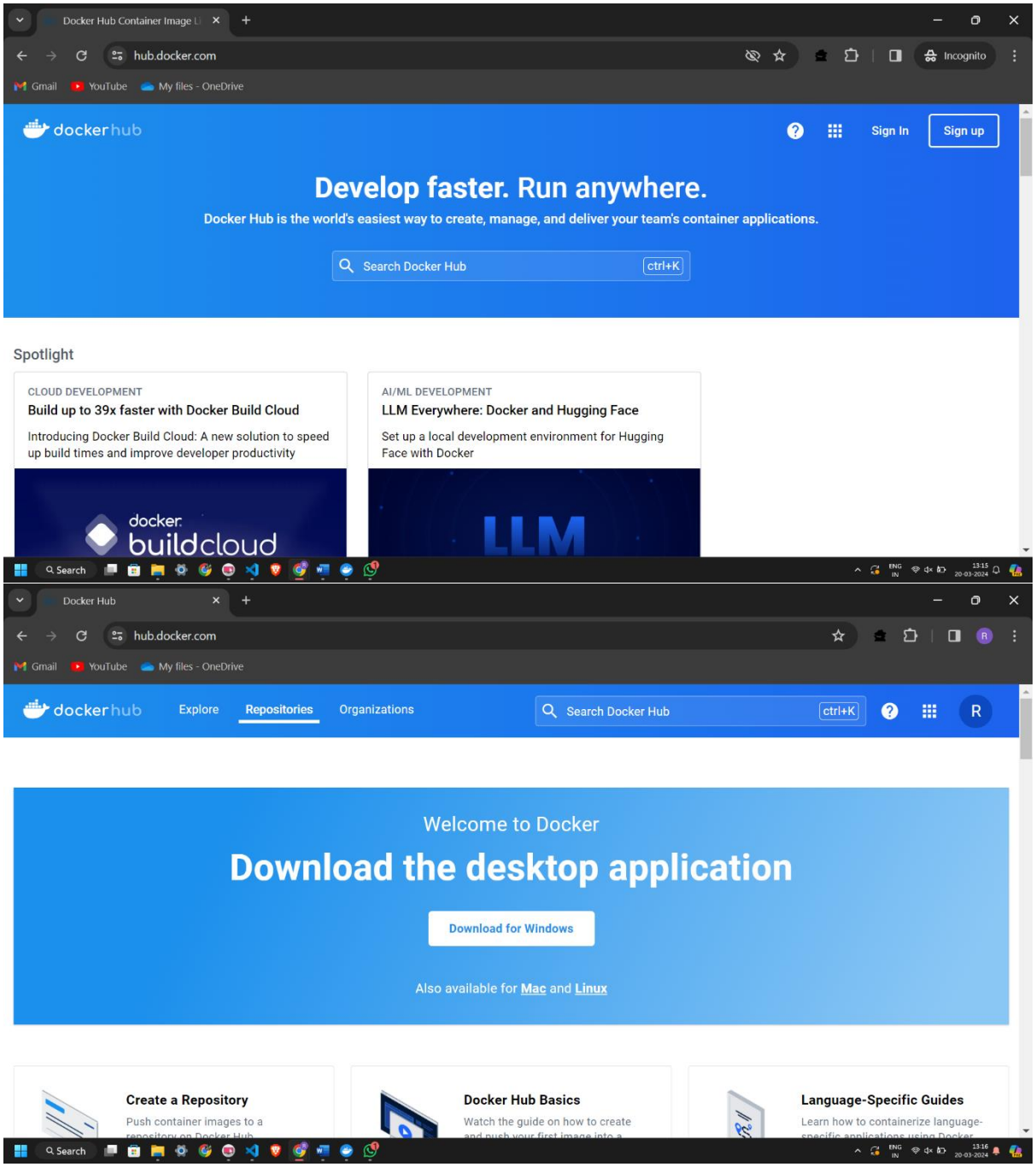
4. Run a container from the built image using the `docker run` command.

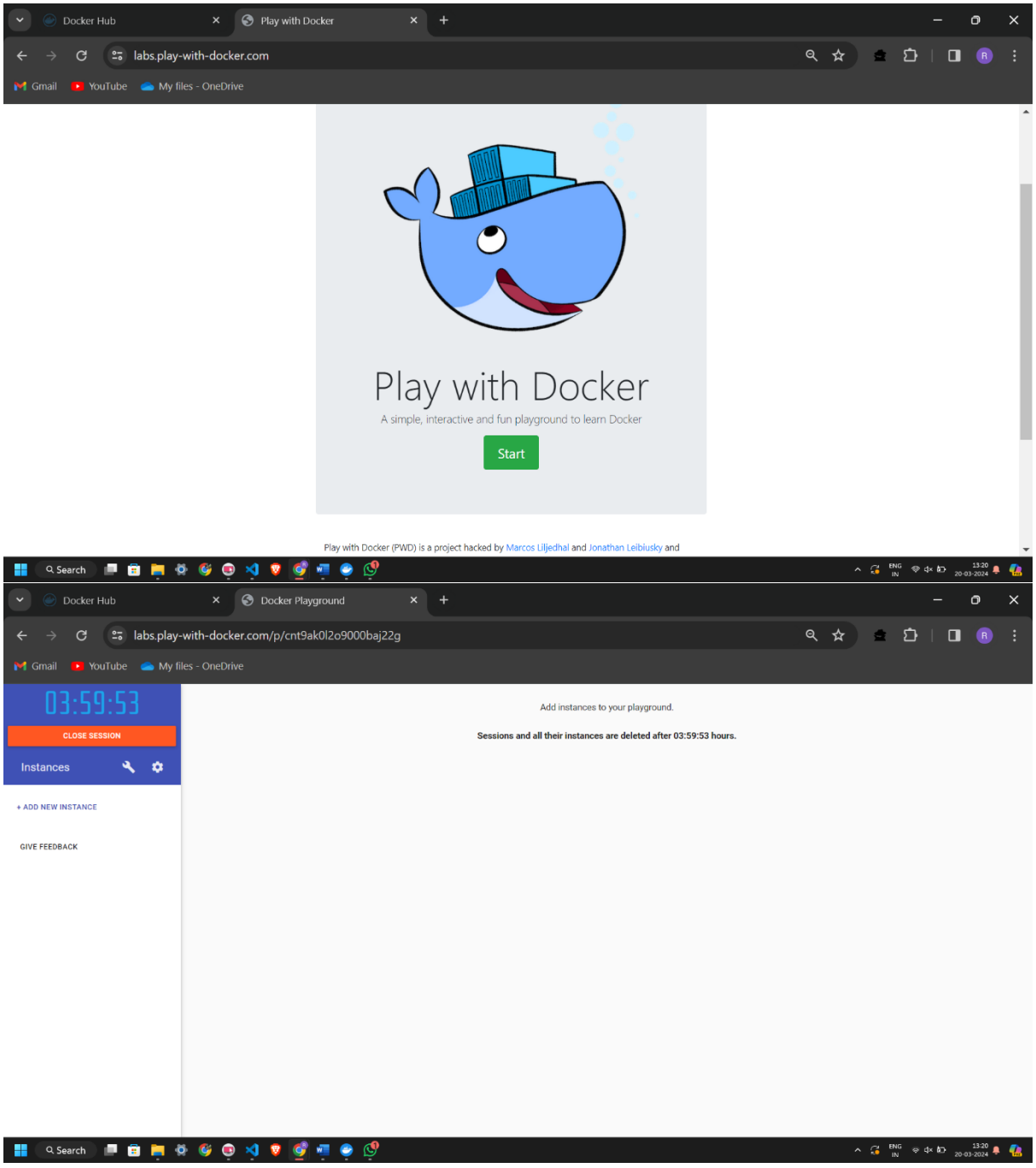
5. Test the application running inside the container.

6. Take screenshots of each step, including the Docker file, building the image, running the container, and testing the application.

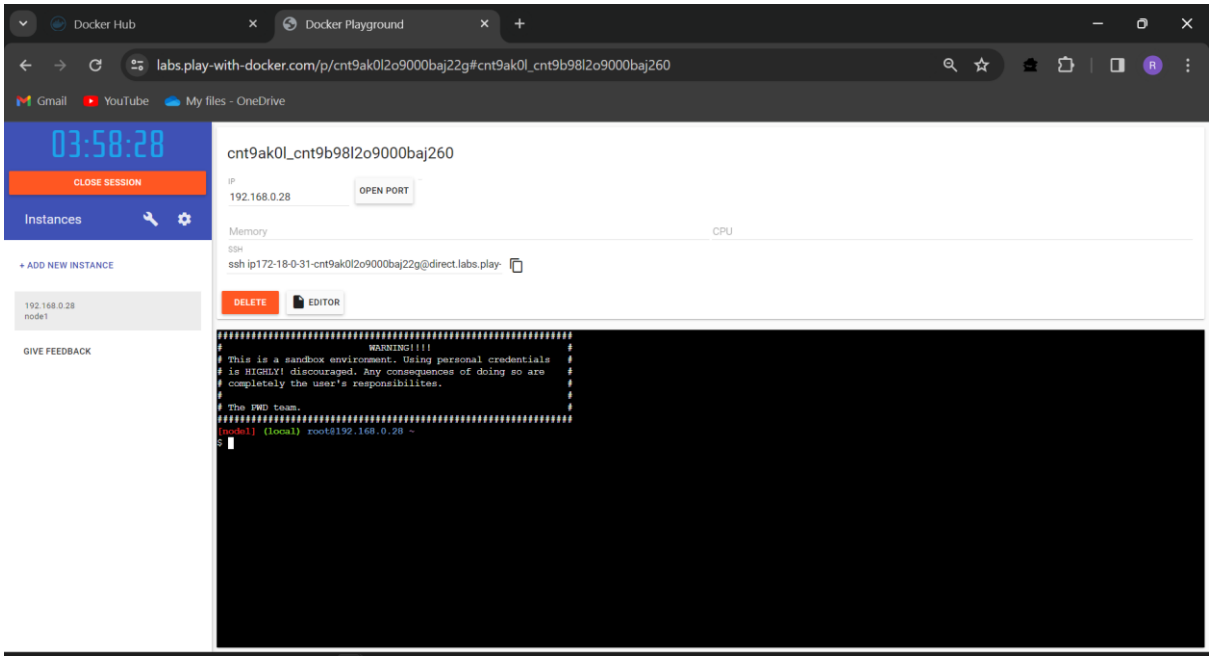
7. Compile the screenshots into a document for submission.

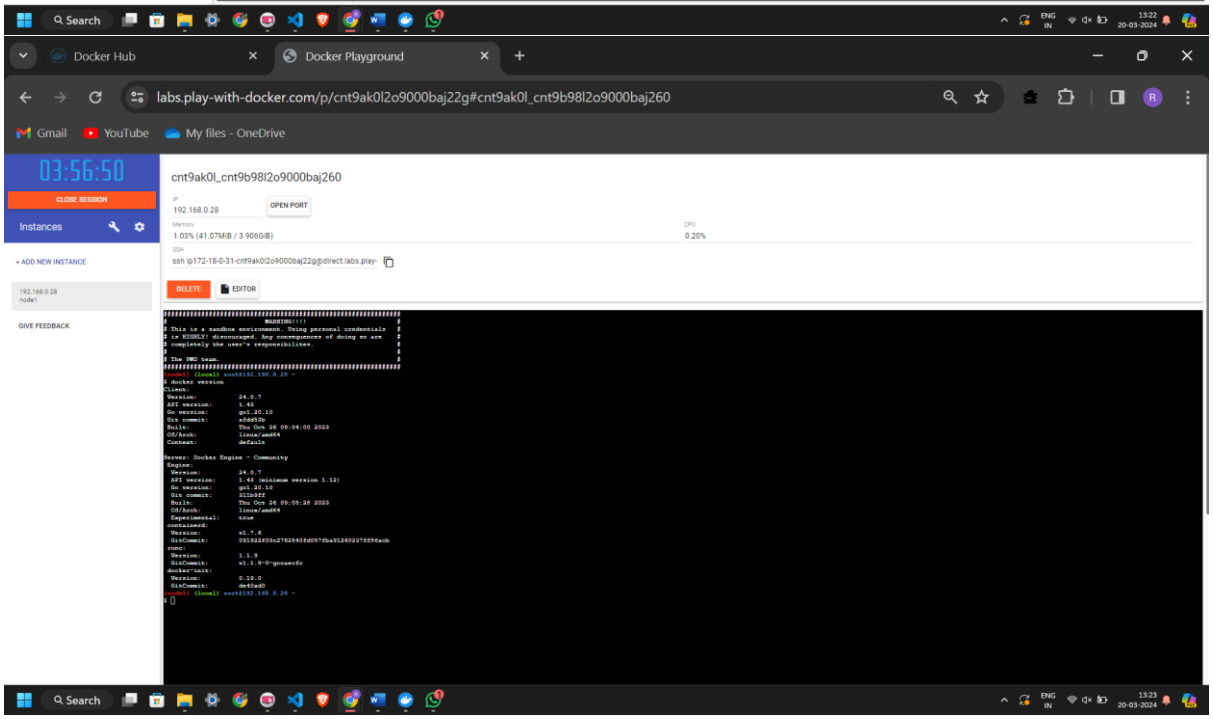
Screenshots:

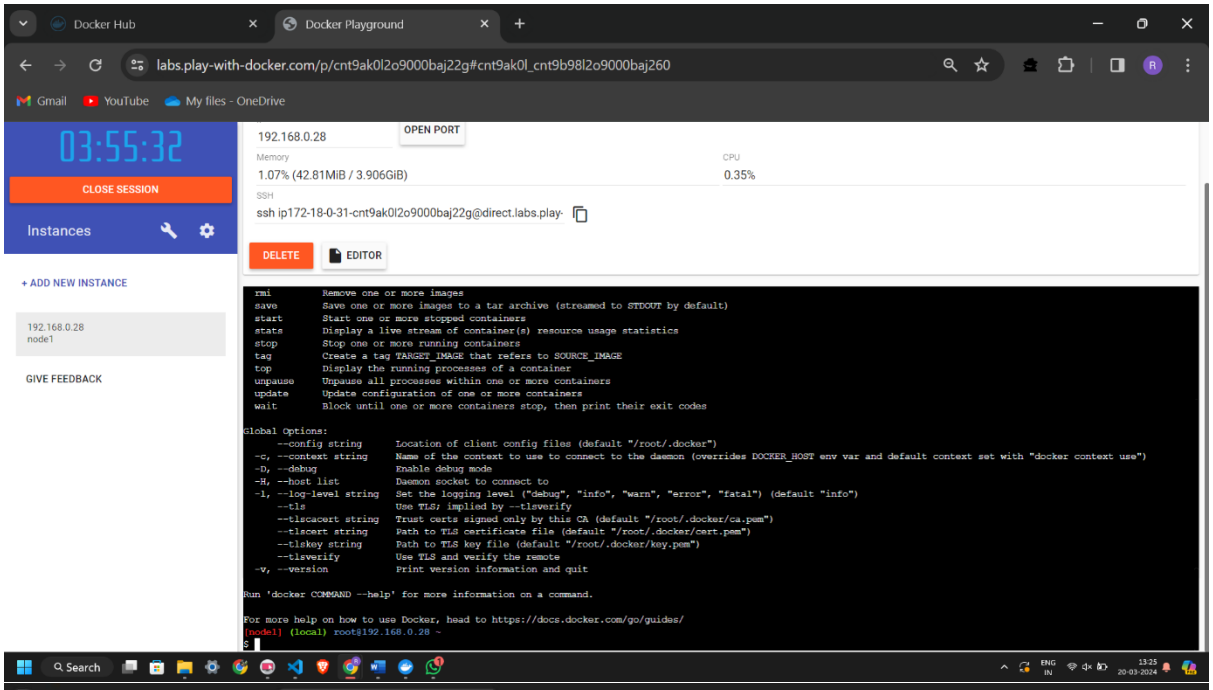

Docker Hub Container Image
hub.docker.com
Incognito
Gmail
YouTube
My files - OneDrive
dockerhub
Sign In
Sign up
Develop faster. Run anywhere.
Docker Hub is the world's easiest way to create, manage, and deliver your team's container applications.
Search Docker Hub
ctrl+K
Spotlight
CLOUD DEVELOPMENT
Build up to 39x faster with Docker Build Cloud
Introducing Docker Build Cloud: A new solution to speed up build times and improve developer productivity
AI/ML DEVELOPMENT
LLM Everywhere: Docker and Hugging Face
Set up a local development environment for Hugging Face with Docker
docker buildcloud
LLM
Search
Docker Hub
hub.docker.com
Gmail
YouTube
My files - OneDrive
dockerhub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
R
Welcome to Docker
Download the desktop application
Download for Windows
Also available for Mac and Linux
Create a Repository
Push container images to a
Docker Hub Basics
Watch the guide on how to create
Language-Specific Guides
Learn how to containerize language-
Search

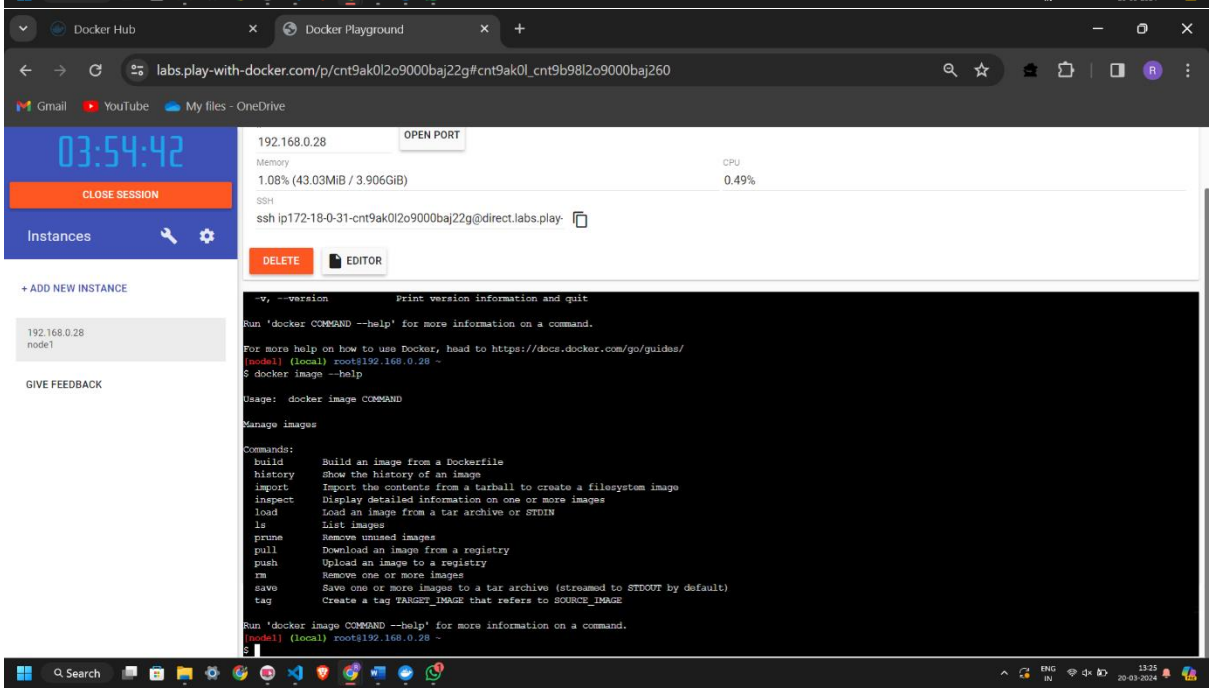

Docker Hub
Play with Docker
labs.play-with-docker.com
Gmail
YouTube
My files - OneDrive
Play with Docker
A simple, interactive and fun playground to learn Docker
Start
Play with Docker (PWD) is a project hacked by Marcos Liljedhal and Jonathan Leibiusky and

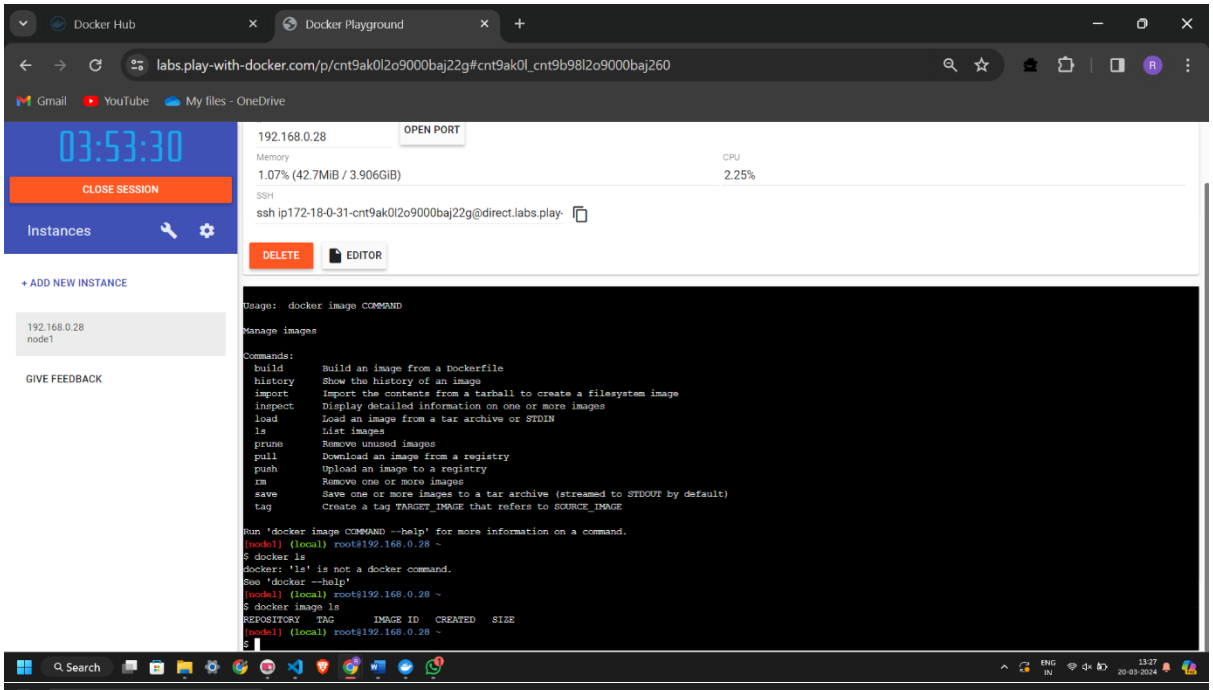
Search
Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g
Gmail
YouTube
My files - OneDrive
03:59:53
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
GIVE FEEDBACK
Add instances to your playground.
Sessions and all their instances are deleted after 03:59:53 hours.
Search

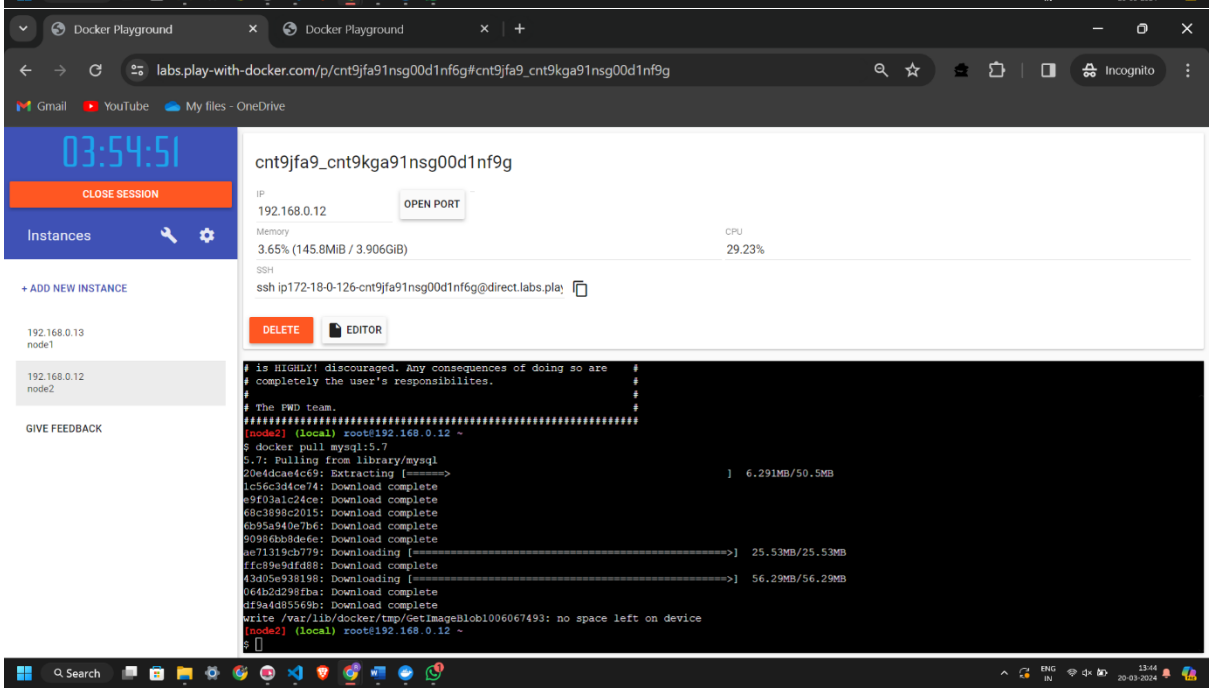

Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g#cnt9ak0l_cnt9b98l2o9000baj260
Gmail
YouTube
My files - OneDrive
03:58:28
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
GIVE FEEDBACK
cnt9ak0l_cnt9b98l2o9000baj260
IP
192.168.0.28
OPEN PORT
Memory
CPU
SSH
ssh ip172-18-0-31-cnt9ak0l2o9000baj22g@direct.labs.play-
DELETE
EDITOR
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@192.168.0.28 ~
$

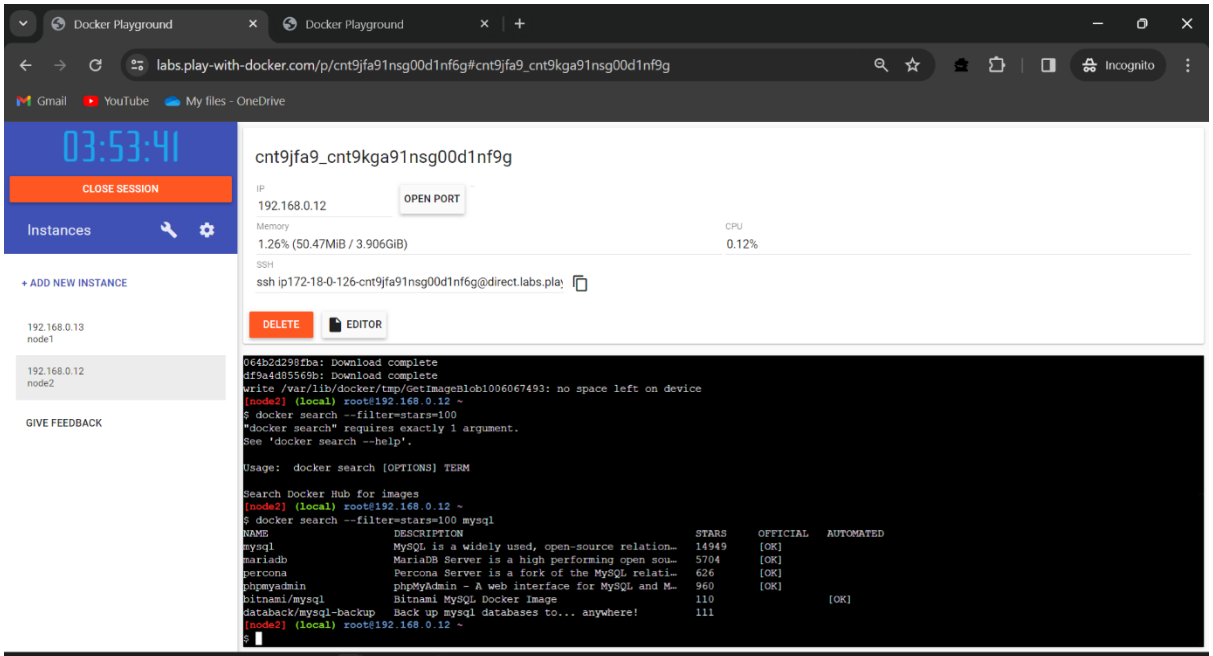

Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g#cnt9ak0l_cnt9b98l2o9000baj260
Gmail
YouTube
My files - OneDrive
03:56:50
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
GIVE FEEDBACK
cnt9ak0l_cnt9b98l2o9000baj260
IP
192.168.0.28
OPEN PORT
Memory
1.03% (41.07MiB / 3.906GiB)
CPU
0.20%
SSH
ssh ip172-18-0-31-cnt9ak0l2o9000baj22g@direct.labs.play-
DELETE
EDITOR
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@192.168.0.28 ~
$ docker version
Client:
Version: 24.0.7
API version: 1.43
Go version: go1.20.10
Git commit: afdd53b
Built: Thu Oct 26 09:04:00 2023
OS/Arch: linux/amd64
Context: default
Server: Docker Engine - Community
Engine:
Version: 24.0.7
API version: 1.43 (minimum version 1.12)
Go version: go1.20.10
Git commit: 311b9ff
Built: Thu Oct 26 09:05:28 2023
OS/Arch: linux/amd64
Experimental: true
containerd:
Version: v1.7.6
GitCommit: 091922f03c2762540fd057fba91260237ff86acb
runc:
Version: 1.1.9
GitCommit: v1.1.9-0-gccaecfc
docker-init:
Version: 0.19.0
GitCommit: de40ad0
[node1] (local) root@192.168.0.28 ~
$

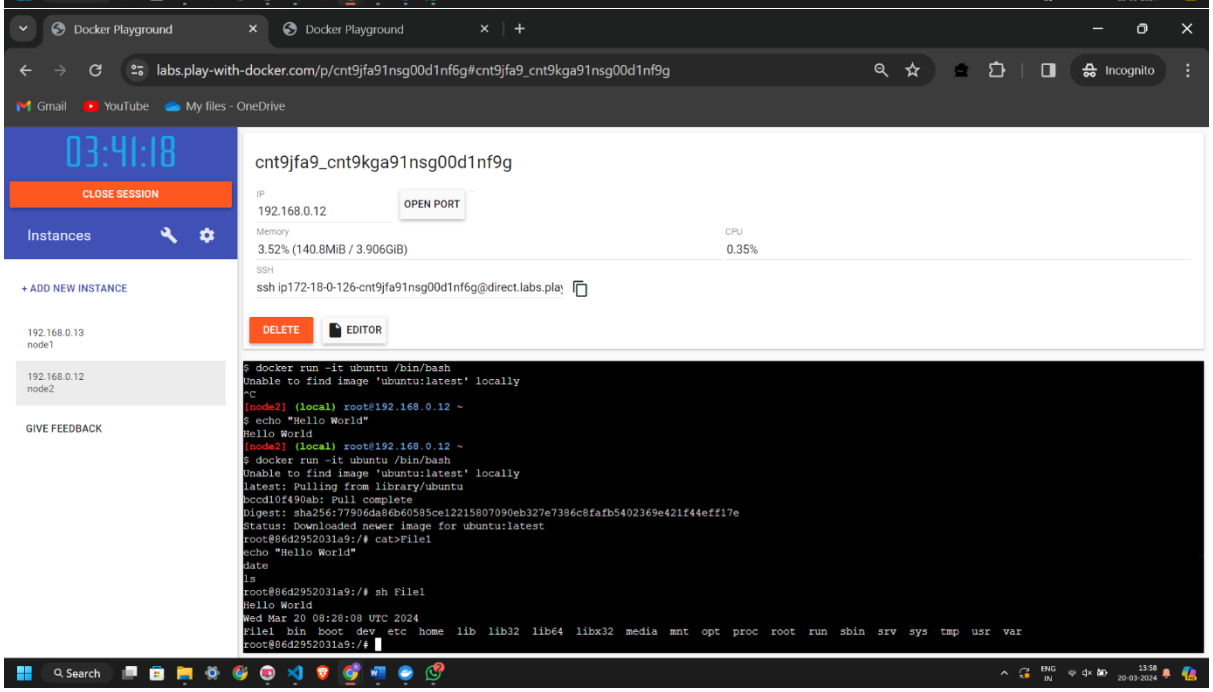

Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g#cnt9ak0l_cnt9b98l2o9000baj260
Gmail
YouTube
My files - OneDrive
03:55:32
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
GIVE FEEDBACK
192.168.0.28
OPEN PORT
Memory
1.07% (42.81MiB / 3.906GiB)
CPU
0.35%
SSH
ssh ip172-18-0-31-cnt9ak0l2o9000baj22g@direct.labs.play-
DELETE
EDITOR
rmi Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
start Start one or more stopped containers
stats Display a live stream of container(s) resource usage statistics
stop Stop one or more running containers
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top Display the running processes of a container
unpause Unpause all processes within one or more containers
update Update configuration of one or more containers
wait Block until one or more containers stop, then print their exit codes
Global Options:
--config string Location of client config files (default "/root/.docker")
-c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var and default context set with "docker context use")
-D, --debug Enable debug mode
-H, --host list Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls Use TLS; implied by --tlsverify
--tlscacert string Trust certs signed only by this CA (default "/root/.docker/ca.pem")
--tlscert string Path to TLS certificate file (default "/root/.docker/cert.pem")
--tlskey string Path to TLS key file (default "/root/.docker/key.pem")
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.28 ~
$
Search

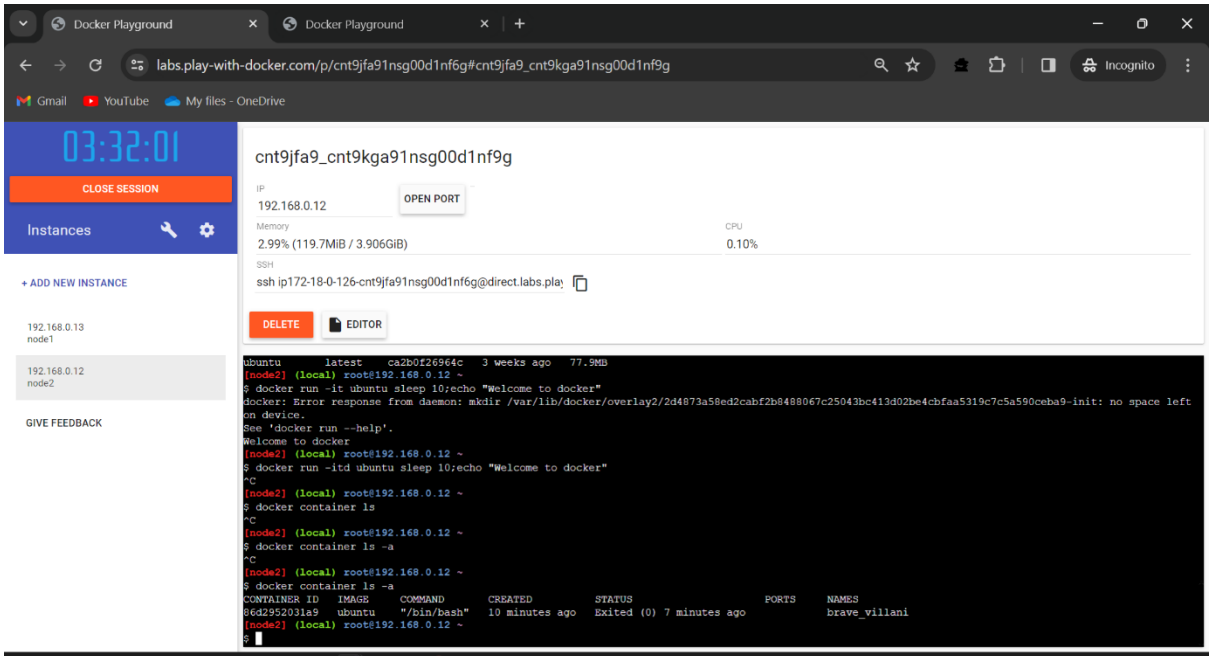

Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g#cnt9ak0l_cnt9b98l2o9000baj260
Gmail
YouTube
My files - OneDrive
03:54:42
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
GIVE FEEDBACK
192.168.0.28
OPEN PORT
Memory
1.08% (43.03MiB / 3.906GiB)
CPU
0.49%
SSH
ssh ip172-18-0-31-cnt9ak0l2o9000baj22g@direct.labs.play-
DELETE
EDITOR
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.28 ~
$ docker image --help
Usage: docker image COMMAND
Manage images
Commands:
build Build an image from a Dockerfile
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Display detailed information on one or more images
load Load an image from a tar archive or STDIN
ls List images
prune Remove unused images
pull Download an image from a registry
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.28 ~
$
Search

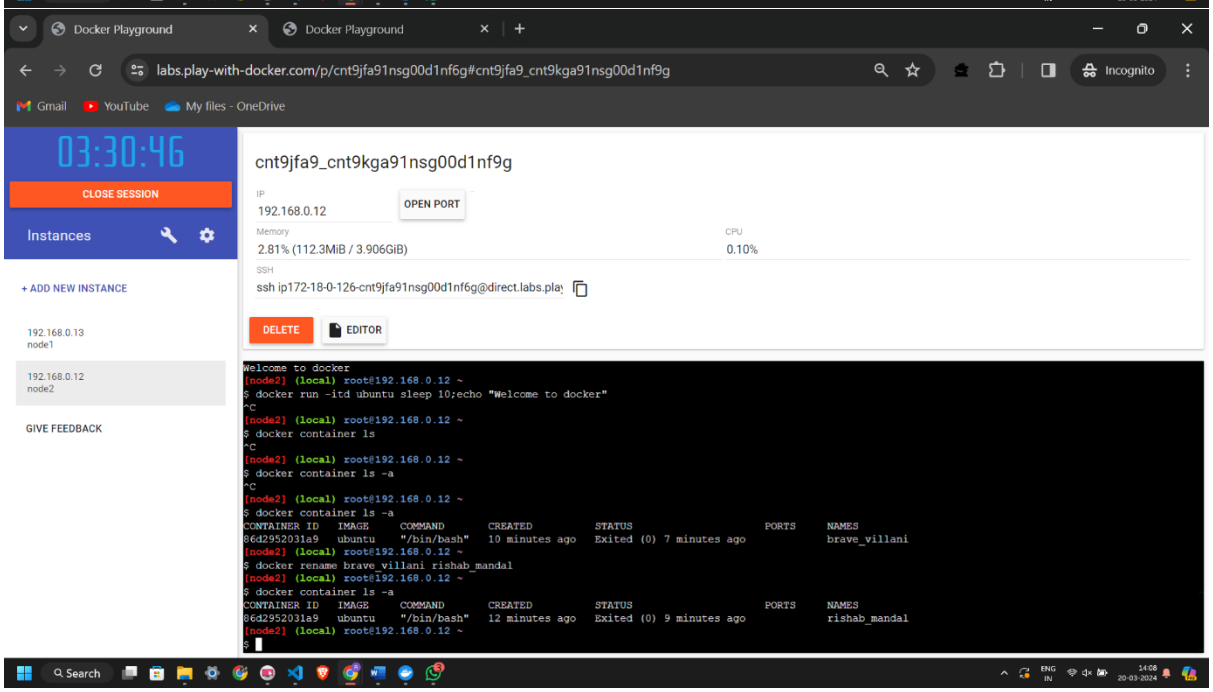

Docker Hub
Docker Playground
labs.play-with-docker.com/p/cnt9ak0l2o9000baj22g#cnt9ak0l_cnt9b98l2o9000baj260
Gmail
YouTube
My files - OneDrive
03:53:30
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.28
node1
GIVE FEEDBACK
192.168.0.28
OPEN PORT
Memory
1.07% (42.7MiB / 3.906GiB)
CPU
2.25%
SSH
ssh ip172-18-0-31-cnt9ak0l2o9000baj22g@direct.labs.play-
DELETE
EDITOR
Usage: docker image COMMAND
Manage images
Commands:
build Build an image from a Dockerfile
history Show the history of an image
import Import the contents from a tarball to create a filesystem image
inspect Display detailed information on one or more images
load Load an image from a tar archive or STDIN
ls List images
prune Remove unused images
pull Download an image from a registry
push Upload an image to a registry
rm Remove one or more images
save Save one or more images to a tar archive (streamed to STDOUT by default)
tag Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
Run 'docker image COMMAND --help' for more information on a command.
[node1] (local) root@192.168.0.28 ~
$ docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
[node1] (local) root@192.168.0.28 ~
$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
[node1] (local) root@192.168.0.28 ~
$
Search

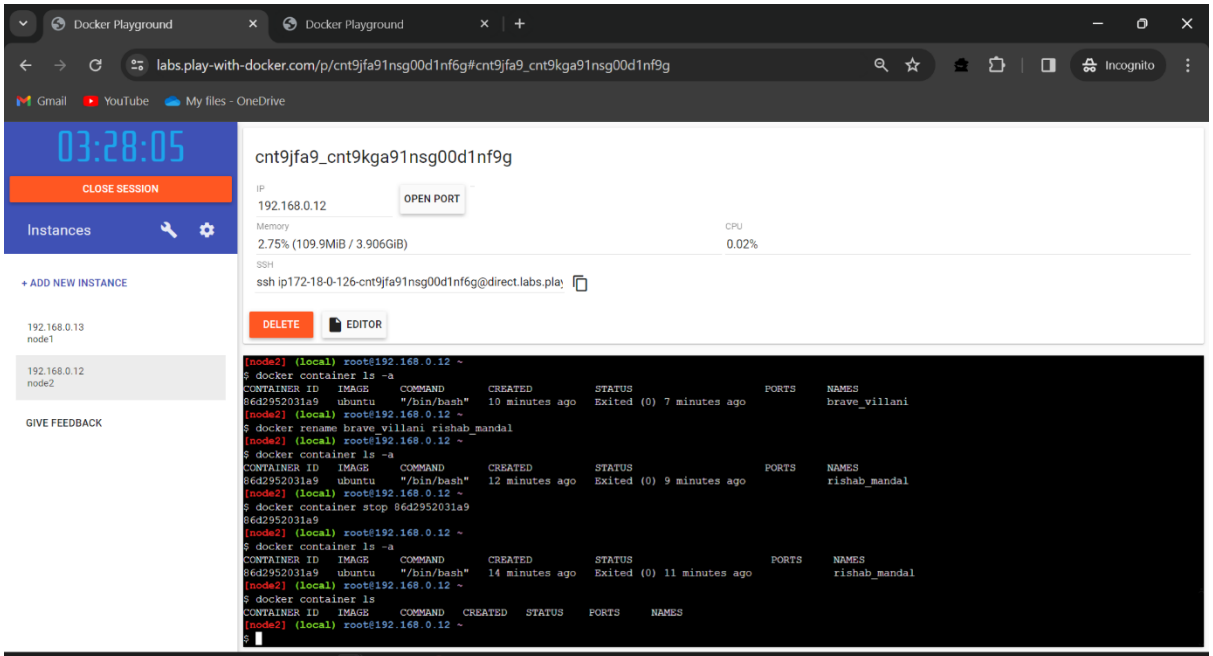

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:54:51
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
3.65% (145.8MiB / 3.906GiB)
CPU
29.23%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
is HIGHLY! discouraged. Any consequences of doing so are
completely the user's responsibilites.
# #
The PWD team.
[node2] (local) root@192.168.0.12 ~
$ docker pull mysql:5.7
5.7: Pulling from library/mysql
20e4dcae4c69: Extracting [======>] 6.291MB/50.5MB
1c56c3d4ce74: Download complete
e9f03a1c24ce: Download complete
68c3898c2015: Download complete
6b95a940e7b6: Download complete
90986bb8de6e: Download complete
ae71319cb779: Downloading [==========>] 25.53MB/25.53MB
ffc89e9dfd88: Download complete
43d05e938198: Downloading [==========>] 56.29MB/56.29MB
064b2d298fba: Download complete
df9a4d85569b: Download complete
write /var/lib/docker/tmp/GetImageBlob1006067493: no space left on device
[node2] (local) root@192.168.0.12 ~
$
Search

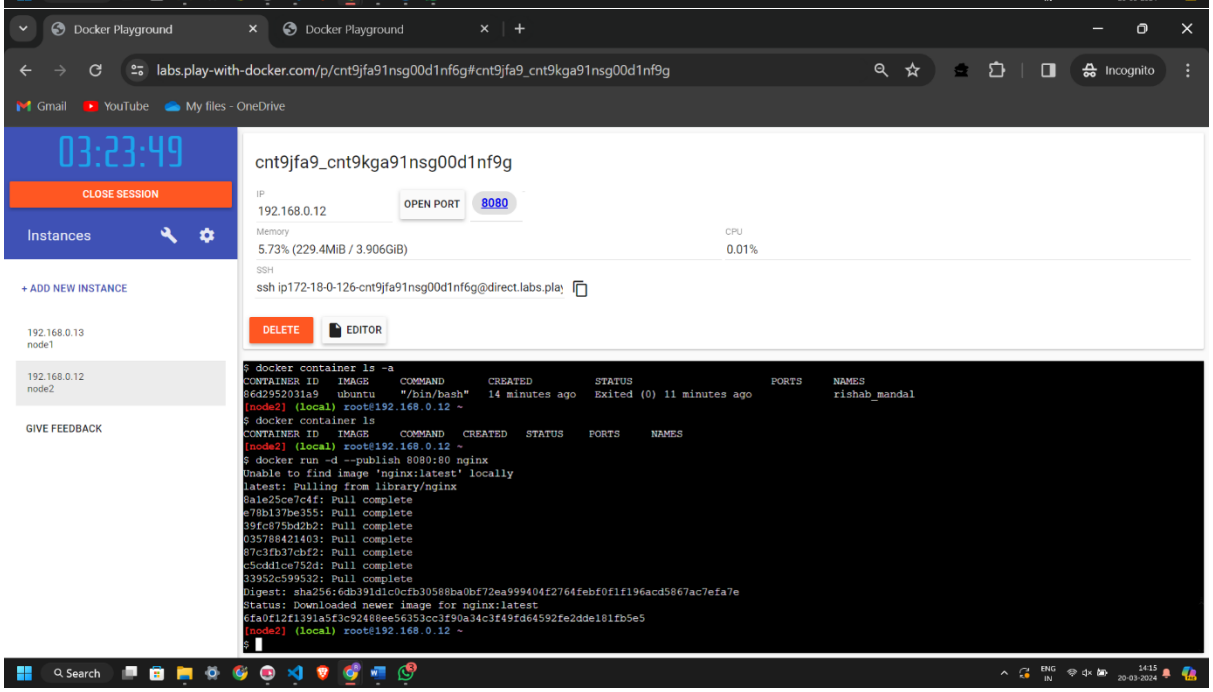

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:53:41
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
1.26% (50.47MiB / 3.906GiB)
CPU
0.12%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
064b2d298fba: Download complete
df9a4d85569b: Download complete
write /var/lib/docker/tmp/GetImageBlob1006067493: no space left on device
[node2] (local) root@192.168.0.12 ~
$ docker search --filter=stars=100
"docker search" requires exactly 1 argument.
See 'docker search --help'.
Usage: docker search [OPTIONS] TERM
Search Docker Hub for images
[node2] (local) root@192.168.0.12 ~
$ docker search --filter=stars=100 mysql
NAME DESCRIPTION STARS OFFICIAL AUTOMATED
mysql MySQL is a widely used, open-source relation… 14949 [OK]
mariadb MariaDB Server is a high performing open sou… 5704 [OK]
percona Percona Server is a fork of the MySQL relati… 626 [OK]
phpmyadmin phpMyAdmin - A web interface for MySQL and M… 960 [OK]
bitnami/mysql Bitnami MySQL Docker Image 110 [OK]
databack/mysql-backup Back up mysql databases to... anywhere! 111
[node2] (local) root@192.168.0.12 ~
$
Search

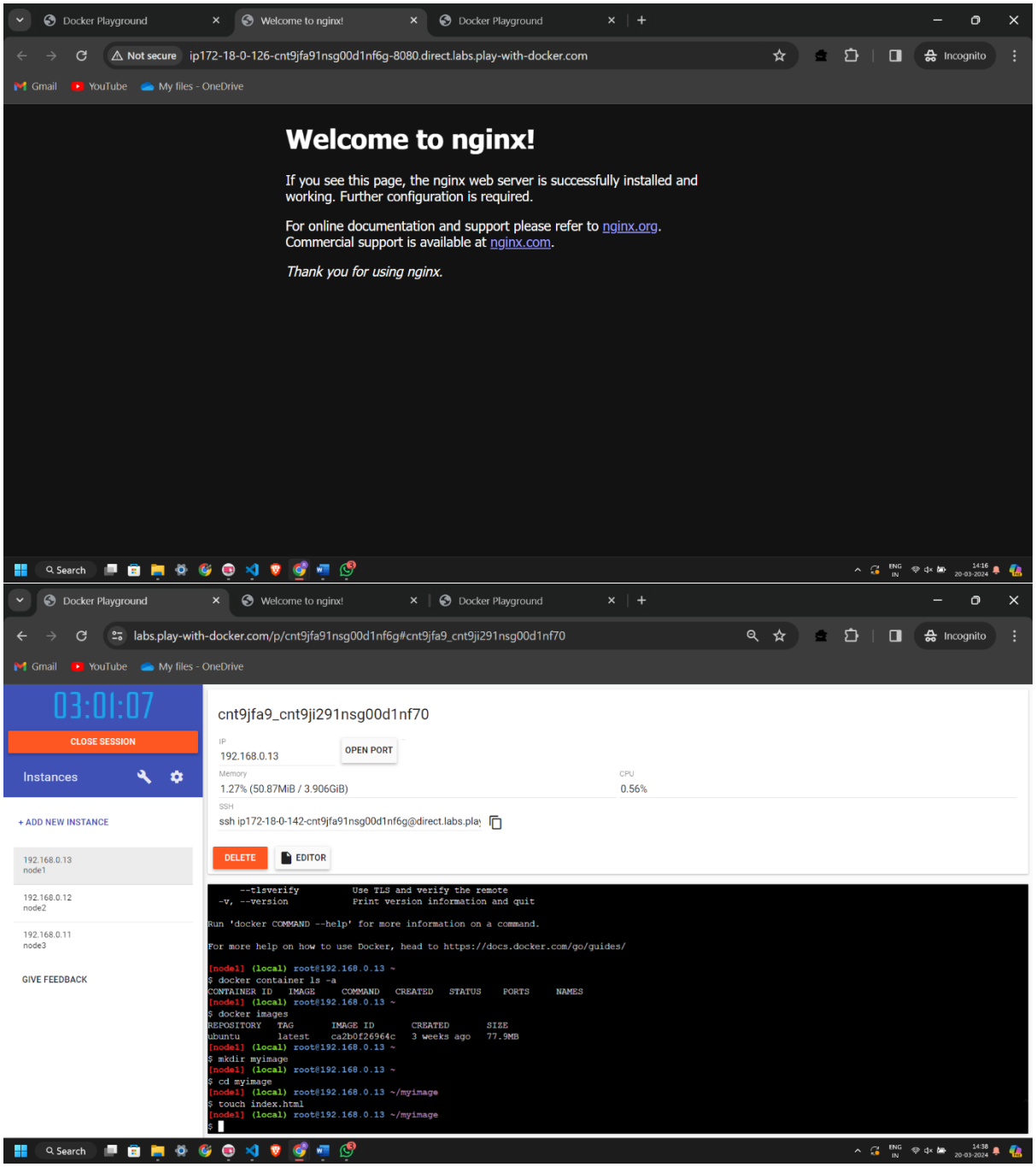

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:41:18
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
3.52% (140.8MiB / 3.906GiB)
CPU
0.35%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
$ docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
^C
[node2] (local) root@192.168.0.12 ~
$ echo "Hello World"
Hello World
[node2] (local) root@192.168.0.12 ~
$ docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@86d2952031a9:/# cat>File1
echo "Hello World"
date
ls
root@86d2952031a9:/# sh File1
Hello World
Wed Mar 20 08:28:08 UTC 2024
File1 bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@86d2952031a9:/#
Search

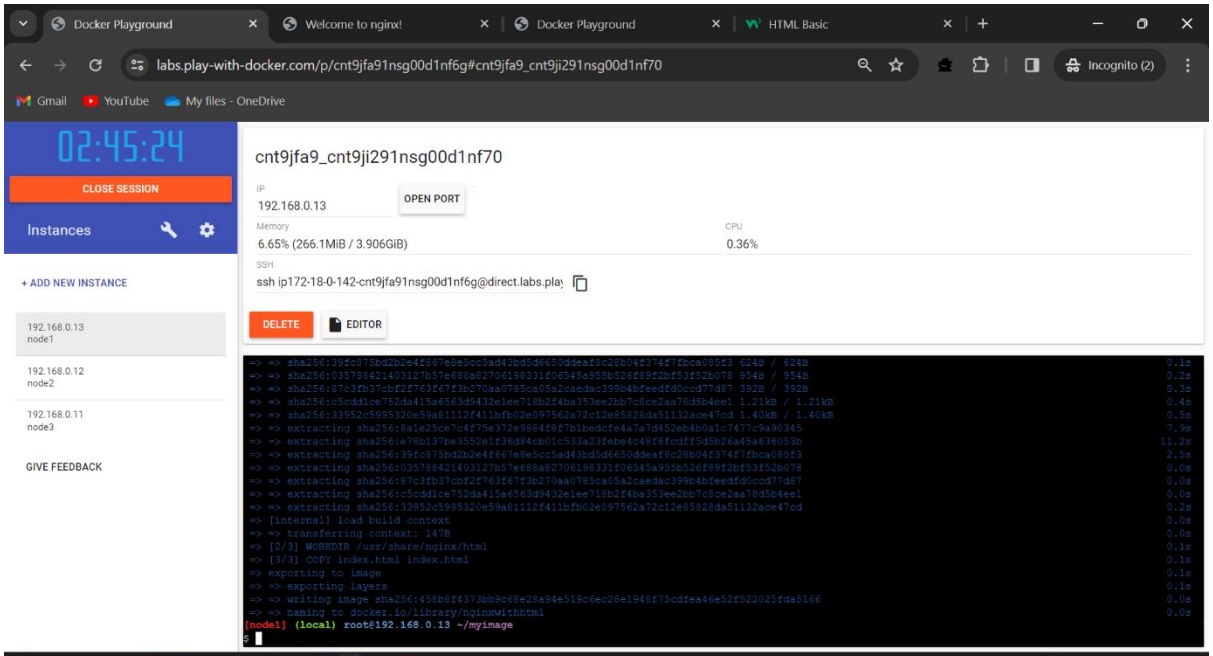

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:32:01
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
2.99% (119.7MiB / 3.906GiB)
CPU
0.10%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
ubuntu latest ca2b0f26964c 3 weeks ago 77.9MB
[node2] (local) root@192.168.0.12 ~
$ docker run -it ubuntu sleep 10;echo "Welcome to docker"
docker: Error response from daemon: mkdir /var/lib/docker/overlay2/2d4873a58ed2cabf2b8488067c25043bc413d02be4cbfaa5319c7c5a590ceba9-init: no space left on device.
See 'docker run --help'.
[node2] (local) root@192.168.0.12 ~
$ docker run -itd ubuntu sleep 10;echo "Welcome to docker"
Welcome to docker
[node2] (local) root@192.168.0.12 ~
$ docker container ls
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 10 minutes ago Exited (0) 7 minutes ago brave_villani
[node2] (local) root@192.168.0.12 ~
$
Search
ENG
IN
14:07
20-03-2024

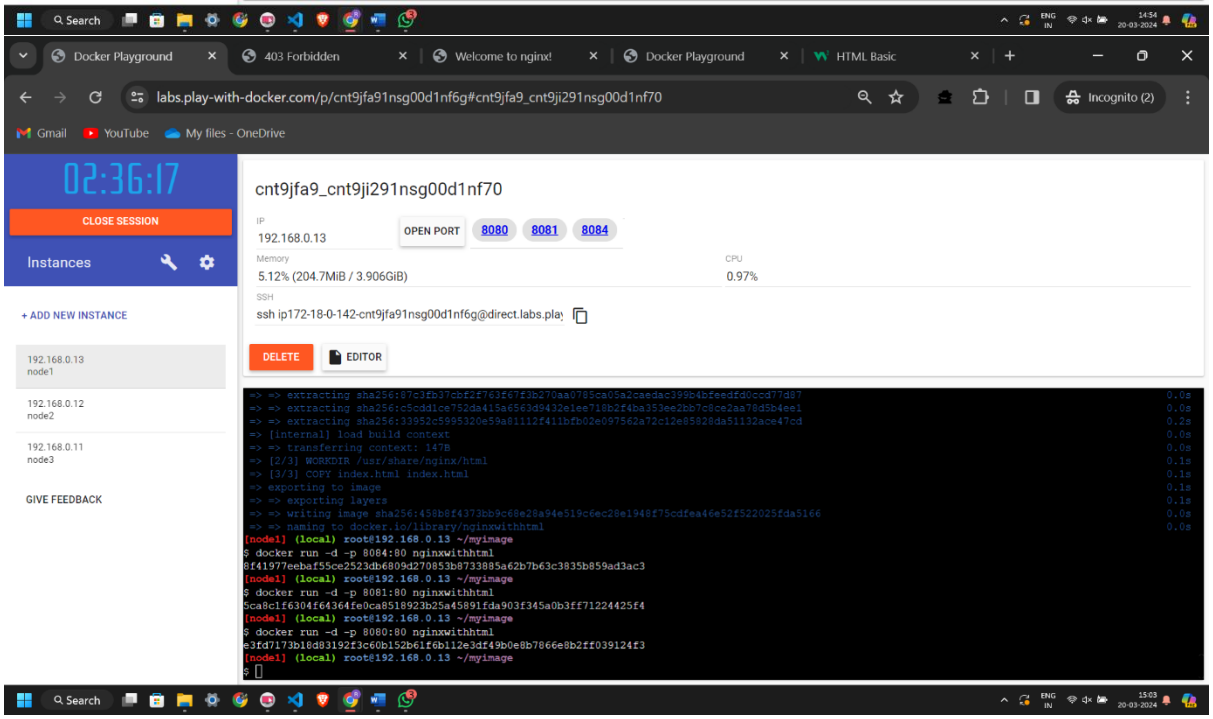

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:30:46
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
2.81% (112.3MiB / 3.906GiB)
CPU
0.10%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
Welcome to docker
[node2] (local) root@192.168.0.12 ~
$ docker run -itd ubuntu sleep 10;echo "Welcome to docker"
[node2] (local) root@192.168.0.12 ~
$ docker container ls
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 10 minutes ago Exited (0) 7 minutes ago brave_villani
[node2] (local) root@192.168.0.12 ~
$ docker rename brave_villani rishab_mandal
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 12 minutes ago Exited (0) 9 minutes ago rishab_mandal
[node2] (local) root@192.168.0.12 ~
$
Search
ENG
IN
14:08
20-03-2024

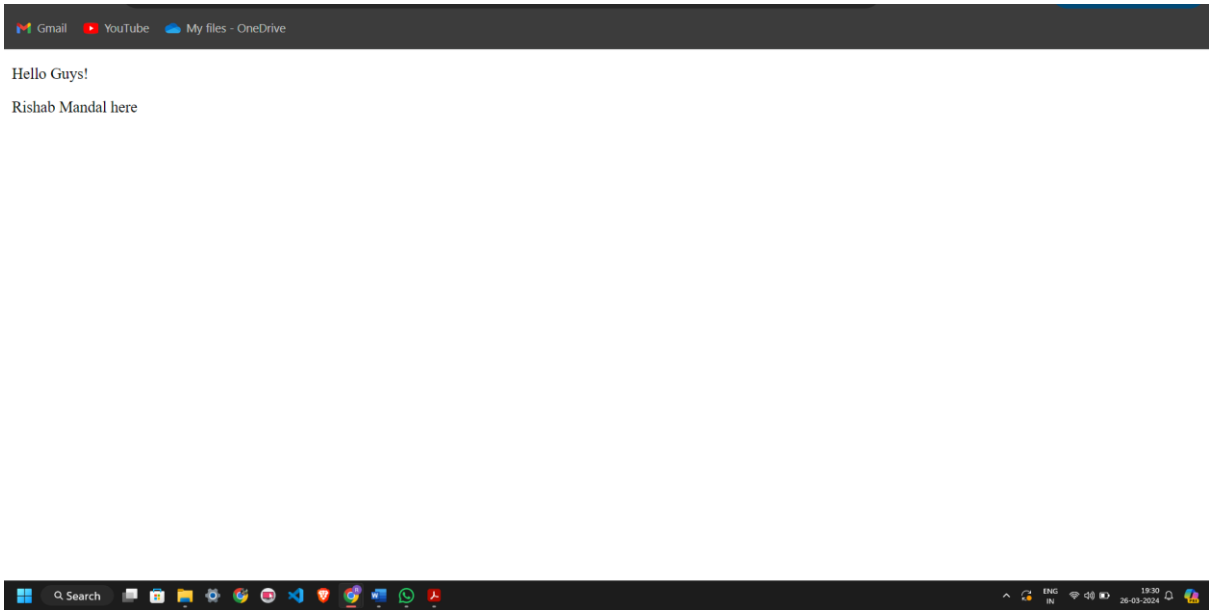

Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:28:05
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
Memory
2.75% (109.9MiB / 3.906GiB)
CPU
0.02%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 10 minutes ago Exited (0) 7 minutes ago brave_villani
[node2] (local) root@192.168.0.12 ~
$ docker rename brave_villani rishab_mandal
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 12 minutes ago Exited (0) 9 minutes ago rishab_mandal
[node2] (local) root@192.168.0.12 ~
$ docker container stop 86d2952031a9
86d2952031a9
[node2] (local) root@192.168.0.12 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 14 minutes ago Exited (0) 11 minutes ago rishab_mandal
[node2] (local) root@192.168.0.12 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node2] (local) root@192.168.0.12 ~
$
Search
ENG
IN
14:11
20-03-2024


Docker Playground
Docker Playground
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9kga91nsg00d1nf9g
Incognito
Gmail
YouTube
My files - OneDrive
03:23:49
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
GIVE FEEDBACK
cnt9jfa9_cnt9kga91nsg00d1nf9g
IP
192.168.0.12
OPEN PORT
8080
Memory
5.73% (229.4MiB / 3.906GiB)
CPU
0.01%
SSH
ssh ip172-18-0-126-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
86d2952031a9 ubuntu "/bin/bash" 14 minutes ago Exited (0) 11 minutes ago rishab_mandal
[node2] (local) root@192.168.0.12 ~
$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node2] (local) root@192.168.0.12 ~
$ docker run -d --publish 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8ale25ce7c4f: Pull complete
e78b137be355: Pull complete
89fc875bd2b2: Pull complete
035788421403: Pull complete
87c3fb37cbf2: Pull complete
c5cdd1ce752d: Pull complete
53952c599532: Pull complete
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
6fa0f12f1391a5f3c92488ee56353cc3f90a34c3f49fd64592fe2dde181fb5e5
[node2] (local) root@192.168.0.12 ~
$
Search
ENG
IN
14:15
20-03-2024


Docker Playground
Welcome to nginx!
Docker Playground
Not secure
ip172-18-0-126-cnt9jfa91nsg00d1nf6g-8080.direct.labs.play-with-docker.com
Incognito
Gmail
YouTube
My files - OneDrive
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.
Search
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9ji291nsg00d1nf70
03:01:07
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
192.168.0.11
node3
GIVE FEEDBACK
cnt9jfa9_cnt9ji291nsg00d1nf70
IP
192.168.0.13
OPEN PORT
Memory
1.27% (50.87MiB / 3.906GiB)
CPU
0.56%
SSH
ssh ip172-18-0-142-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
--tlsverify Use TLS and verify the remote
-v, --version Print version information and quit
Run 'docker COMMAND --help' for more information on a command.
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
[node1] (local) root@192.168.0.13 ~
$ docker container ls -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
[node1] (local) root@192.168.0.13 ~
$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest ca2b0f26964c 3 weeks ago 77.9MB
[node1] (local) root@192.168.0.13 ~
$ mkdir myimage
[node1] (local) root@192.168.0.13 ~
$ cd myimage
[node1] (local) root@192.168.0.13 ~/myimage
$ touch index.html
[node1] (local) root@192.168.0.13 ~/myimage
$


Docker Playground
Welcome to nginx!
Docker Playground
HTML Basic
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9ji291nsg00d1nf70
Incognito (2)
Gmail
YouTube
My files - OneDrive
02:45:24
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
192.168.0.11
node3
GIVE FEEDBACK
cnt9jfa9_cnt9ji291nsg00d1nf70
IP
192.168.0.13
OPEN PORT
Memory
6.65% (266.1MiB / 3.906GiB)
CPU
0.36%
SSH
ssh ip172-18-0-142-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
[node1] (local) root@192.168.0.13 ~/myimage
$


Docker Playground
403 Forbidden
Welcome to nginx!
Docker Playground
HTML Basic
labs.play-with-docker.com/p/cnt9jfa91nsg00d1nf6g#cnt9jfa9_cnt9ji291nsg00d1nf70
Incognito (2)
Gmail
YouTube
My files - OneDrive
02:36:17
CLOSE SESSION
Instances
+ ADD NEW INSTANCE
192.168.0.13
node1
192.168.0.12
node2
192.168.0.11
node3
GIVE FEEDBACK
cnt9jfa9_cnt9ji291nsg00d1nf70
IP
192.168.0.13
OPEN PORT
8080
8081
8084
Memory
5.12% (204.7MiB / 3.906GiB)
CPU
0.97%
SSH
ssh ip172-18-0-142-cnt9jfa91nsg00d1nf6g@direct.labs.play
DELETE
EDITOR
[node1] (local) root@192.168.0.13 ~/myimage
$ docker run -d -p 8084:80 nginxwithhtml
8f41977eebaf55ce2523db6809d270853b8733885a62b7b63c3835b859ad3ac3
[node1] (local) root@192.168.0.13 ~/myimage
$ docker run -d -p 8081:80 nginxwithhtml
5ca8c1f630f4f6436fe0ca8518923b25a45891fda903f345a0b3ff71224425f4
[node1] (local) root@192.168.0.13 ~/myimage
$ docker run -d -p 8080:80 nginxwithhtml
e3fd7173b18d83192f3c60b152b61f6b112e3df49b0e8b7866e8b2ff039124f3
[node1] (local) root@192.168.0.13 ~/myimage
$



Conclusion:

In conclusion, Docker containerization emerges as a transformative technology, offering developers a versatile platform for building, packaging, and deploying applications with enhanced efficiency, portability, and scalability, revolutionizing modern software development practices.