

Docker Commands

Docker is a containerization platform that allows applications to run in isolated environments.

1 Install Docker

```
bash
CopyEdit
sudo apt install docker.io -y
```

- Installs Docker on an Ubuntu system.

2 Restart Docker Service

```
bash
CopyEdit
sudo service docker restart
```

- Ensures Docker starts properly after installation.

3 Check Docker Status

```
bash
CopyEdit
sudo service docker status
```

- Verifies whether Docker is running.

4 Add User to Docker Group

```
bash
CopyEdit
sudo usermod -aG docker $USER
```

- Allows the current user to run Docker without sudo.

5 Verify Docker Installation

```
bash
CopyEdit
docker images # Lists available Docker images
docker ps      # Shows running containers
```

6 Fix Permission Issues

```
bash
CopyEdit
sudo chmod 666 /var/run/docker.sock
```

- Grants non-root users access to the Docker socket.

Docker Compose Commands

Docker Compose is used to manage multi-container Docker applications using a YAML configuration file.

1 Install Docker Compose

```
bash
CopyEdit
sudo apt install docker-compose -y
```

- Installs Docker Compose.

2 Download the Latest Version

```
bash
CopyEdit
```

```
sudo curl -L  
"https://github.com/docker/compose/releases/latest/download/docker-compose-\$\(uname -s\)-\$\(uname -m\)" -o /usr/local/bin/docker-compose
```

- Fetches the latest stable version.

3 Make Docker Compose Executable

```
bash  
CopyEdit  
sudo chmod +x /usr/local/bin/docker-compose
```

- Grants execution permissions.

4 Verify Installation

```
bash  
CopyEdit  
docker-compose --version
```

- Confirms Docker Compose is installed.

5 Example docker-compose.yml File

```
yaml  
CopyEdit  
version: '3'  
services:  
  web:  
    image: nginx:latest  
    ports:  
      - "80:80"  
  db:  
    image: mysql:latest  
    environment:  
      - MYSQL_ROOT_PASSWORD=secret
```

- Defines an Nginx web server and a MySQL database.

6 Start Services

```
bash
CopyEdit
docker-compose up -d
```

- Starts containers in detached mode.

7 Execute a Shell Inside a Running Container

```
bash
CopyEdit
docker exec -it <container_name> /bin/bash
```

- Accesses a running container's shell.

8 Access MySQL Database

```
bash
CopyEdit
mysql -u root -p
```

- Logs into MySQL inside the container.

Kubernetes Commands

Kubernetes (k8s) is an orchestration system for automating deployment, scaling, and management of containerized applications.

1 Download kubectl

```
bash
```

CopyEdit

```
curl -LO https://dl.k8s.io/release/v1.32.0/bin/linux/amd64/kubectl
```

- Downloads the kubectl binary.

2 Install kubectl

bash

CopyEdit

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

- Installs kubectl with proper permissions.

3 Make kubectl Executable

bash

CopyEdit

```
chmod +x kubectl
```

- Ensures kubectl has execution rights.

4 Move kubectl to Local Bin

bash

CopyEdit

```
mkdir -p ~/.local/bin
```

```
mv ./kubectl ~/.local/bin/kubectl
```

- Places kubectl in the user's local binary path.

5 Verify kubectl Installation

bash

CopyEdit

```
kubectl version --client
```

- Checks if `kubectl` is installed correctly.



Jenkins is an open-source automation tool used for **Continuous Integration (CI)** and **Continuous Delivery (CD)**.

Why Use Jenkins?

- Automates **builds, testing, and deployment** of applications.
- Supports **plugins** to integrate with various development tools.
- Provides a **web-based interface** for managing pipelines.

```

arulmurugan-19@Arul: ~ + ~
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4c372bbf5de5 mysql:latest "docker-entrypoint.s..." 47 seconds ago Up 45 seconds 3306/tcp, 33060/tcp arulmurugan-19-db-1
arulmurugan-19@Arul:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4c372bbf5de5 mysql:latest "docker-entrypoint.s..." 54 seconds ago Up 52 seconds 3306/tcp, 33060/tcp arulmurugan-19-db-1
b89aaafadad1e nginx:latest "/docker-entrypoint..." 54 seconds ago Created
arulmurugan-19-web-1 b3ac6a5bf19b my-nginx "/docker-entrypoint..." 49 minutes ago Exited (0) 9 minutes ago
nginxx-containe a5784ed12d7e hello-world "/hello" 55 minutes ago Exited (0) 55 minutes ago
quirky_greider
arulmurugan-19@Arul:~$ docker exec -it arulmurugan-19-db-1 /bin/bash
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

USD/INR -0.38%

localhost:8080/job/DevOps%20Day-2/4/

Jenkins

Dashboard > DevOps Day-2 > #4

Status #4 (Mar 22, 2025, 5:39:36 AM) Add description Keep this build forever

Started by user Arul Murugan S Started 37 sec ago Took 34 sec

This run spent:

- 16 ms waiting;
- 34 sec build duration;
- 34 sec total from scheduled to completion.

git Revision: 37bcf3f23e9a15e23a57704bcb856d85bb4ee3b4
Repository: <https://github.com/arulmurugan000/app.git>
refs/remotes/origin/main

</> No changes.

Changes
Console Output
Edit Build Information
Delete build '#4'
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build

30°C Mostly sunny

localhost:8080/job/DevOps%20Day-2/4/console

Console Output

```

Started by user Arul Murugan S
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/DevOps Day-2
[Pipeline] {
[Pipeline] stage
[Pipeline] { (scm)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/DevOps Day-2/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/arulmurugan000/app.git # timeout=10
Fetching upstream changes from https://github.com/arulmurugan000/app.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/arulmurugan000/app.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/main{commit} # timeout=10
Checking out Revision 37bcf3f23e9a15e23a57704bcb856d85bb4ee3b4 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 37bcf3f23e9a15e23a57704bcb856d85bb4ee3b4 # timeout=10

```

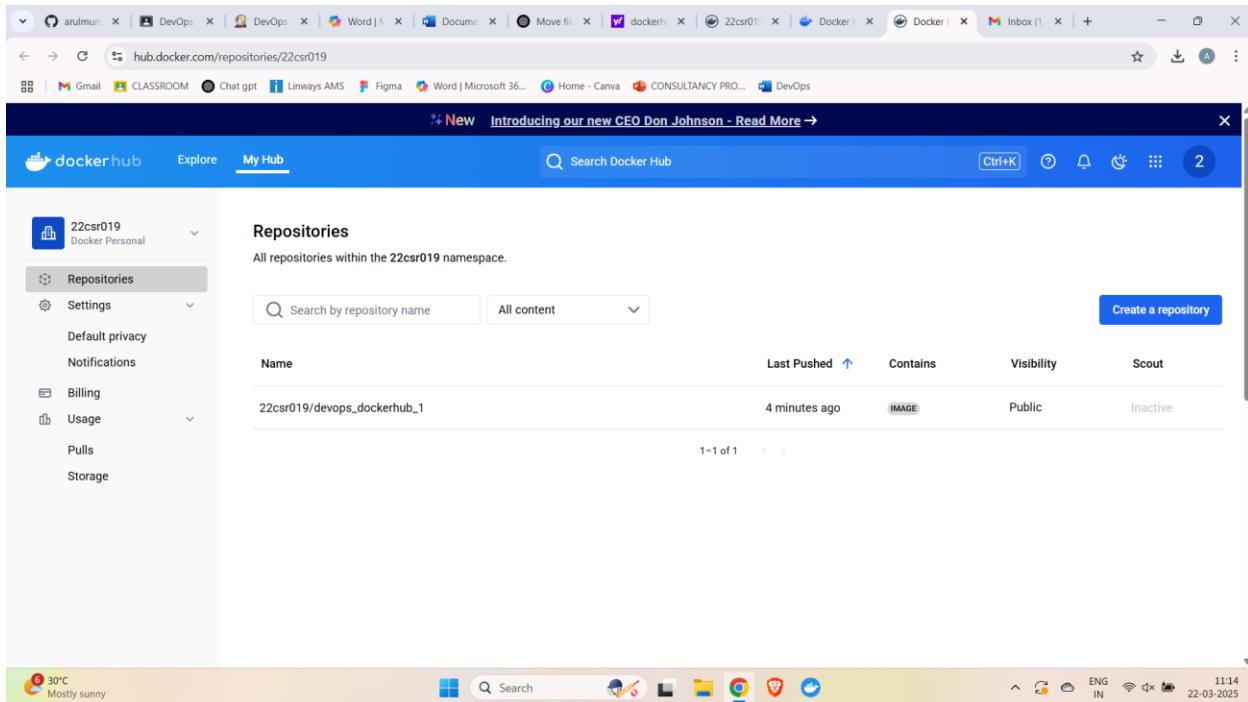
localhost:8080/job/DevOps%20Day-2/4/console

```

3359bc3d7a6a: Waiting
4b7c01ed0534: Waiting
43c9f8a1dd61: Mounted from library/tomcat
4e5b5ab7345: Mounted from library/tomcat
5f70bf18a086: Mounted from library/tomcat
bc05267c613b: Mounted from library/tomcat
f6cad70e3ed0: Pushed
3359bc3d7a6a: Mounted from library/tomcat
39cf0ac89a5a: Mounted from library/tomcat
f844dcf94898: Mounted from library/tomcat
4b7c01ed0534: Mounted from library/tomcat
latest: digest: sha256:9eadce796f05b673e22fa280f3e76e2492066f2bbad1d6cd8210f02dcba10e0f size: 2409
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```





```
pipeline { agent any

stages {
    stage('SCM') {
        steps {
            git branch: 'main', url:
'https://github.com/arulmurugan000/app.git'
        }
    }

    stage('Build') {
        steps {
            sh "mvn clean"
            sh "mvn install"
        }
    }

    stage('Build to Images') {
        steps {
            script {
                sh 'docker build -t 22csr019/devops_dockerhub_1 .'
            }
        }
    }
}
```

```
        }
    }

stage('Push to Hub') {
    steps {
        script {
            docker.withRegistry('https://index.docker.io/v1/',
'docker-credential') {
                sh 'docker push 22csr019/devops_dockerhub_1'
            }
        }
    }
}

}
```