

## DOCKER COMPOSE

Docker Compose is a tool that allows you to define and manage multi-container Docker applications. It simplifies the process of running multiple containers, their configurations, and their interdependencies. Compose uses a YAML file to define the services, networks, and volumes required for your application.

- ❓ Docker Compose is a tool which is used to manage multi container-based applications.
- ❓ Using Docker Compose we can easily setup & deploy multi container-based applications.
- ❓ We will give containers information to Docker Compose using YML file (docker-compose.yml)
- ❓ Docker Compose YML should have all the information related to containers creation.
- ❓ Docker Compose YML File Looks Like:

download docker compose

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

or

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

## JENKINS

Jenkins is an open-source automation tool written in Java programming language that allows continuous integration. Jenkins offers a straightforward way to set up a continuous integration or continuous delivery environment for almost any combination of languages and source code repositories using pipelines, as well as automating other routine development tasks.

The following are the main or most popular Jenkins use cases:

- Continuous Integration: With Jenkins pipelines, we can achieve CI for both applications and infrastructure as code.
- Continuous Delivery: You can set up well-defined and automated application delivery workflows with Jenkins pipelines.

Jenkins achieves CI (Continuous Integration) and CD (Continuous Deployment) with the help of plugins. Plugins are used to allow the integration of various DevOps stages. If you want to integrate a particular tool, you must install the plugins for that tool.

#### ADVANTAGES OF JENKINS:

- It is an open-source tool.
- It is free of cost.
- It does not require additional installations or components. Means it is easy to install.
- Easily configurable.
- It supports 1000 or more plugins to ease your work. If a plugin does not exist, you can write the script for it and share with community.
- It is built in java and hence it is portable.
- It is platform independent. It is available for all platforms and different operating systems. Like OS X, Windows, or Linux.
- Easy support since its open source and widely used.
- Jenkins also supports cloud-based architecture so that we can deploy Jenkins in cloud-based platforms.

#### Commands:

# Install Java and FontConfig

```
sudo apt install fontconfig openjdk-17-jre -y
```

```
java -version
```

# Install Jenkins

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
```

```
echo "deb http://pkg.jenkins.io/debian-stable binary/" | sudo tee /etc/apt/sources.list.d/jenkins.list
```

```
sudo apt update
```

```
sudo apt install jenkins -y
```

```
sudo service jenkins restart
```

```
sudo service jenkins status
```

# Install Docker

```
sudo apt install docker.io -y
sudo service docker restart
sudo service docker status
sudo usermod -aG docker $USER
```

# Verify Docker installation

```
docker images
```

```
docker ps
```

# Fix Docker permissions

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

# Install kubectl

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

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

```
chmod +x kubectl
```

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

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

```
kubectl version --client
```

# Install Minikube

```
curl -LO https://github.com/kubernetes/minikube/releases/latest/download/minikube-linux-amd64
```

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube && rm minikube-linux-amd64
```

# Start Minikube

```
minikube start
```

```
minikube status
```

# Verify Kubernetes setup

```
kubectl get pod
```

```
kubectl get deploy
```

```
kubectl get replica
```

```
kubectl get pod -o wide
```

```
# Docker Compose configuration example
```

```
echo "version: '3'
```

```
services:
```

```
  web:
```

```
    image: nginx:latest
```

```
    ports:
```

```
      - 80:80
```

```
  db:
```

```
    image: mysql:latest
```

```
    environment:
```

```
      - MYSQL_ROOT_PASSWORD=secret" > docker-compose.yml
```

```
# Run Docker Compose
```

```
docker-compose up -d
```

```
# Enter MySQL container
```

```
docker exec -it david-db-1 /bin/bash
```

```
mysql -u root -p
```

```
# Install Docker Compose
```

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

```
# Download latest Docker Compose version
```

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

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

## docker-compose --version

```
nithesh@NITHESH-MP: ~  
nithesh@NITHESH-MP:~$ minikube start  
🐳 minikube v1.35.0 on Ubuntu 22.04 (amd64)  
🔧 Using the docker driver based on existing profile  
👉 Starting "minikube" primary control-plane node in "minikube" cluster  
📡 Pulling base image v0.0.46 ...  
🔄 Restarting existing docker container for "minikube" ...  
❗ Failing to connect to https://registry.k8s.io/ from inside the minikube container  
💡 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/  
📦 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...  
🔍 Verifying Kubernetes components...  
▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5  
🌟 Enabled addons: storage-provisioner, default-storageclass  
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
nithesh@NITHESH-MP:~$ kubectl get pods  
NAME                READY   STATUS    RESTARTS   AGE  
my-replicaset-ct7lh 0/1     Error     0           22h  
my-replicaset-dr9gm 0/1     Error     0           22h  
my-replicaset-dwfps 0/1     Error     0           22h  
my-replicaset-h4zcr 0/1     ImagePullBackOff  0           22h  
my-replicaset-pg9d6 0/1     ErrImagePull  0           22h  
nithesh@NITHESH-MP:~$ kubectl get node  
NAME      STATUS   ROLES    AGE   VERSION  
minikube  Ready    control-plane  23h   v1.32.0  
nithesh@NITHESH-MP:~$
```