**Types of Docker valumes:**

Anonymous Volumes:

docker run -it --name <container\_name> -v /<directory\_name> <image\_name> /bin/bash

Named volume:

docker run -it --name <container\_name> -v <directory\_name>:/<volume \_name> <image\_name> /bin/bash

Host volumes:

docker run -it --name <container\_name> -v <directory\_name>:/<volume \_name> <image\_name> /bin/bash

for checking volumes: docker volume ls

once volume got created directly you are going to inside the container

if you want to see your you have come out of your container by using "exit"

**Minikube install:**

Minikube Installation On Ubuntu Server

Step 1: Update Your System

Ensure your system is up to date by running:

sudo apt update && sudo apt upgrade -y

Step 2: Install Dependencies

Minikube requires a hypervisor to run Kubernetes. You can use KVM, VirtualBox, or other supported hypervisors. For simplicity, this guide uses VirtualBox.

1.Install VirtualBox:

sudo apt install -y virtualbox virtualbox-ext-pack

2. Install conntrack package:

Minikube requires the conntrack package, which you can install using:

sudo apt install -y conntrack

Step 3: Install Minikube

1.Download the Minikube binary:

Visit the Minikube releases page or use the following command to download the latest version:

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

2. Install the Minikube binary:

Move the downloaded binary to /usr/local/bin and make it executable:

sudo install minikube-linux-amd64 /usr/local/bin/minikube

Step 4: Install kubectl

kubectl is the command-line tool for interacting with Kubernetes clusters.

1. Download kubectl:

curl -LO https://dl.k8s.io/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

2.Install kubectl:

Move the binary to /usr/local/bin and make it executable:

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

Step 5. Create a Non-Root User for Minikube

1.Create a New User:

sudo adduser minikube-user

sudo usermod -aG sudo minikube-user

2. Switch to the New User:

su - minikube-user

Step 6. Install Docker:

sudo apt-get update

sudo apt-get install -y docker.io

sudo usermod -aG docker $USER

newgrp docker

Start Minikube with the docker Driver:

minikube start --driver=docker

#Check the status of minikube by using below commands

minikube status

kubectl get nodes

**# How to create namespace**

vi <file\_name.yml>

apiVersion: v1

kind: Namespace

metadata:

name: wipro-ns

kubectl get ns (To see namespaces)

kubectl apply -f <file\_name> (To create any object)

**# How to create pod**

vi <file\_name.yml>

apiVersion: v1

kind: Pod

metadata:

name: wipro-pod

spec:

containers:

- name: demo-1

image: nginx:1.14.2

ports:

- containerPort: 80

**# How to create replicaset**

vi <file\_name.yml>

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: wipro-replicaset

spec:

replicas: 5

selector:

matchLabels:

app: wipro

template:

metadata:

name: new-replica

labels:

app: wipro

spec:

containers:

- name: arjun

image: nginx:1.14.2

**# How to create deployment**

vi <file\_name.yml>

apiVersion: apps/v1

kind: Deployment

metadata:

name: wipro-deployment

spec:

replicas: 3

selector:

matchLabels:

app: nabeel

template:

metadata:

name: new-deployment

labels:

app: nabeel

spec:

containers:

- name: sravani

image: nginx:1.14.2

ports:

- containerPort: 80

**# How to create service yaml file**

vi <file\_name.yml>

apiVersion: v1

kind: Service

metadata:

name: wipro-service

spec:

selector:

app: nabeel

ports:

- protocol: TCP

port: 80

targetPort: 80

type: LoadBalancer

**# How to create ingress**

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: minimal-ingress

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

ingressClassName: nginx-example

rules:

- http:

paths:

- path: /testpath

pathType: Prefix

backend:

service:

name: wipro-service

port:

number: 80

sonarqube install

docker run -d --name sonar -p 9000:9000 sonarqube:lts-community