Docker Compose & CI/CD Pipeline Execution – Detailed Notes

1. Introduction to Docker Compose

Docker Compose is a tool for defining and running multi-container Docker applications. With a single YAML file (`docker-compose.yml`), you can define services, networks, and volumes.

---

2. Installing Docker Compose

Before using Docker Compose, ensure that Docker is installed.
Steps to install Docker Compose:

For Linux/macOS:
Download Docker Compose:
bash
Copy code

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

1.

Give executable permissions:
bash
Copy code

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

2.

Verify installation:
bash
Copy code

```
docker-compose --version
```

Output should display something like:
pgsql
Copy code

```
docker-compose version 1.29.2, build 5becea4c
```

3.

For Windows:

- Docker Compose comes pre-installed with **Docker Desktop**.

---

3. Creating a Docker Compose File

A `docker-compose.yml` file defines multiple services (like a database and a web app) to run together.

Example: Running a MySQL & Web Application

Create a `docker-compose.yml` file:

yaml
Copy code

```yaml
version: "3.8"

services:
  db:
    image: mysql:latest
    container_name: ananth_db_1
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: user
      MYSQL_PASSWORD: userpassword
    ports:
      - "3306:3306"
    networks:
      - mynetwork

  web:
    image: my-web-app
    container_name: ananth_web_1
    depends_on:
      - db
    ports:
      - "8080:80"
```

```
    networks:
      - mynetwork

networks:
  mynetwork:
    driver: bridge
```

This setup: ✅ Starts a **MySQL** container with predefined credentials.
✅ Starts a **Web App** container, which connects to MySQL.
✅ Both services communicate via a **Docker network**.

---

4. Running Docker Compose

Navigate to the directory containing `docker-compose.yml`

bash
Copy code

```
cd ~/devops_tasks/task1
```

1.

Start the containers in detached mode:

bash
Copy code

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

2.

Check running containers:

bash
Copy code

```
docker ps
```

3.

Stop and remove containers:

bash
Copy code

```
docker-compose down
```

4.

---

5. Running MySQL inside a Container

Once the database is running, you can access it:

bash
Copy code

```
docker exec -it ananth_db_1 /bin/bash
mysql -u root -p
```

💡 This allows interaction with the MySQL server.

---

6. CI/CD Pipeline Execution (Jenkins)

A **CI/CD pipeline** automates the build, test, and deployment process. Your screenshot shows a successful Jenkins pipeline execution.

Basic Jenkins Pipeline for Docker Build & Deployment

1. **Install Jenkins Plugins:**
    - Docker Pipeline
    - Git Plugin
2. **Create a Jenkinsfile:**

groovy
Copy code

```
pipeline {
    agent any
    environment {
        DOCKER_IMAGE = 'my-web-app'
    }
    stages {
        stage('Checkout') {
            steps {
                git 'https://github.com/your-repo.git'
            }
        }
        stage('Build Docker Image') {
            steps {
```

```
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }
        stage('Push to Docker Hub') {
            steps {
                withDockerRegistry([credentialsId:
'docker-hub-credentials']) {
                    sh 'docker push
my-dockerhub-user/$DOCKER_IMAGE:latest'
                }
            }
        }
        stage('Deploy') {
            steps {
                sh 'docker-compose up -d'
            }
        }
    }
}
```

3. **Trigger the Pipeline:**
    - Configure **Jenkins** to pull the latest code and run the pipeline.
    - The pipeline will: ✅ **Build** the Docker image
      ✅ **Push** the image to Docker Hub
      ✅ **Deploy** using Docker Compose

```
ananth@DESKTOP-LQBQG2D:~$ docker start ananth_db_1 ananth_web_1 minikube
ananth_db_1
ananth_web_1
minikube
ananth@DESKTOP-LQBQG2D:~$ docker exec -it ananth_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>
```

```
3359bc3d7a6a: Mounted from library/tomcat
f844dcf94898: Mounted from library/tomcat
4b7c01ed0534: Mounted from library/tomcat
latest: digest: sha256:1e7a64d4f2b24daf27d5154454704d07790c826c99f18266a1cb586f61d8f873 size: 2409
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```