Practical 7

Aim: Setting Up CI/CD with Jenkins:

1. Prerequisites
   - **VS Code**
   - Make sure Docker is running

2. Create a Jenkins project folder

```
mkdir jenkins-docker
cd jenkins-docker
```

- Create **docker-compose.yml**:

```
version: '3.8'

services:
  jenkins:
    image: jenkins/jenkins:lts
    container_name: jenkins
    restart: unless-stopped
    ports:
      - "8080:8080"   # Jenkins UI
      - "50000:50000" # For Jenkins agents
    volumes:
      - jenkins_home:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock  # so Jenkins can use host Docker
volumes:
  jenkins_home:
```

3. **Start Jenkins**

```
docker compose up -d
```

Check:

```
docker ps
```

Open in browser : http://localhost:8080

In UserName: Admin

Get the admin password:(On VSCODE Terminal by running command):

```
docker exec jenkins cat /var/jenkins_home/secrets/initialAdminPassword
```

Unlock Jenkins → Install **Suggested Plugins** → Create Admin User.

Part 2: Prepare Your Flask/FastAPI App

Example structure:

```
myapp/
│── app.py
│── requirements.txt
│── Dockerfile
│── tests/
│   └── test_sample.py
│── Jenkinsfile
```

app.py (Flask example)

```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def home():
    return "Hello from Flask + Jenkins!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

requirements.txt

```
flask
pytest
```

Dockerfile

```dockerfile
FROM python:3.10-slim

WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .
CMD ["python", "app.py"]
```

tests/test_sample.py

```python
def test_example():
    assert 2 + 2 == 4
```

Part 3: Jenkins Pipeline (CI/CD)

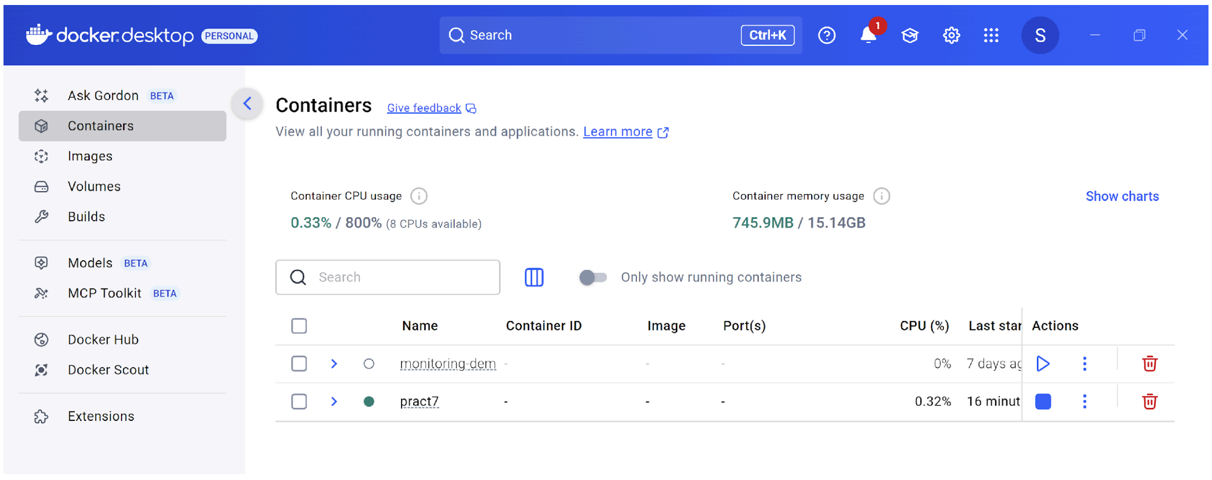
Create a **Jenkinsfile** in your repo:

```groovy
pipeline {
    agent any

    environment {
        DOCKER_IMAGE = "myflaskapi:latest"
        CONTAINER_NAME = "flaskapi-container"
    }

    stages {
        stage('Checkout') {
            steps {
                git branch: 'main', url: 'https://github.com/your-repo/myapp.git'
            }
        }

        stage('Build') {
            steps {
                sh 'docker build -t $DOCKER_IMAGE .'
            }
        }

        stage('Test') {
            steps {
                sh 'docker run --rm $DOCKER_IMAGE pytest'
            }
        }

        stage('Deploy') {
            steps {
                sh 'docker rm -f $CONTAINER_NAME || true'
                sh 'docker run -d --name $CONTAINER_NAME -p 5000:5000
$DOCKER_IMAGE'
            }
        }
    }

    post {
        success {
            echo "🚀 App deployed at http://localhost:5000"
        }
        failure {
            echo "❌ Build/Test/Deploy failed"
        }
    }
}
```

🔹 **Part 4: Create Pipeline Job in Jenkins**

1. Go to Jenkins → **New Item** → Name: FlaskAPI-CI-CD.
2. Select **Pipeline** → OK.
3. In **Pipeline > Definition**, choose:
   - *Pipeline script from SCM*
   - SCM: **Git**
   - Repo URL: your GitHub repo URL
   - Script Path: Jenkinsfile
4. Save → Build Now.

- **Part 5: Test Deployment in Docker**



Containers Give feedback

View all your running containers and applications. Learn more

Container CPU usage: 0.33% / 800% (8 CPUs available)

Container memory usage: 745.9MB / 15.14GB

Name	Container ID	Image	Port(s)	CPU (%)	Last star	Actions
monitoring-dem	-	-	-	0%	7 days ag	
pract7	-	-	-	0.32%	16 minut	