**Docker Installation & Container Management Assignment**

**Objective:**

This assignment will help you install Docker on an Ubuntu system, manage containers, and build a Python Flask application inside a Docker container.

**Part 1: Install Docker on Ubuntu**

**Step 1: Install Docker & Docker Compose**

1. **Update the package list**

sudo apt update

1. **Install Docker**

sudo apt install docker.io -y

1. **Start and Enable Docker Service**

sudo systemctl start docker

sudo systemctl enable docker

1. **Install Docker Compose**

sudo curl -L "https://github.com/docker/compose/releases/download/v2.14.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

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

1. **Verify Docker Compose Installation**

docker-compose --version

1. **Add Current User to Docker Group** (Run this and restart the system)

sudo usermod -aG docker $USER

**Setup Dockerhub Account**

[**https://hub.docker.com/**](https://hub.docker.com/)

**Part 2: Docker Basics**

**Run the following commands and document the output:**

1. **Check Docker version:**

docker --version

1. **View Docker system information:**

docker info

1. **Login to Docker Hub (Optional)**

docker login

1. **List running containers:**

docker ps

1. **Pull the Nginx image:**

docker pull nginx

1. **Run an Nginx container:**

docker run -p 8080:80 nginx

1. **View logs of the running container:**

docker logs -f <container\_id>

1. **List all containers (including stopped ones):**

docker ps -a

1. **Stop and remove a container:**

docker kill <container\_id>

**Part 3: Container Management**

**Run and document the following commands:**

1. **List running containers:**

docker container ls

1. **List all containers:**

docker container ls -a

1. **Stop a specific container:**

docker container stop <container\_id>

1. **Start a stopped container:**

docker container start <container\_id>

1. **Remove a container:**

docker container rm <container\_id>

1. **Access an Nginx container shell:**

docker exec -it nginx /bin/bash

1. **Run Jenkins container:**

docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins

1. **Run Jenkins in detached mode:**

docker run -d -p 8080:8080 -p 50000:50000 jenkins/jenkins

1. **Run Jenkins with named volume:**

docker run --name jenkins -d -p 8081:8080 -p 50001:50000 -v jenkins:/var/jenkins\_home jenkins/jenkins:lts

1. **Run container with resource limits:**

docker run -m 1GB -e JAVA\_OPTS="-Xms500M -Xmx500M" -p 8080:8080 -p 50000:50000 jenkins/jenkins

1. **Run Nginx container with memory and CPU limits:**

docker run -d -p 8080:80 --memory="1g" --cpus="1.0" nginx

1. **Monitor container resource usage:**

docker stats

**Part 4: Working with Docker Volumes**

1. **Create a named volume:**

docker volume create nginx\_logs

1. **List available volumes:**

docker volume ls

1. **Inspect a volume:**

docker volume inspect nginx\_logs

1. **Run a container with a volume mounted:**

docker run -d --name my-nginx -p 8080:80 -v my\_nginx\_data:/usr/share/nginx/html nginx

**Part 5: Build and Run a Python Flask Application in Docker**

**Step 1: Create the Application Files**

**app.py**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello():

return "Hello from Flask!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=5000)

**requirements.txt**

flask

redis

**Step 2: Run the Python Application Locally**

1. Install Flask
2. pip install flask
3. Run the application
4. python3 app.py

**Step 3: Create a Dockerfile**

**Dockerfile**

FROM python:3.9-slim

# Set the working directory in the container

WORKDIR /app

# Copy the current directory contents into the container at /app

COPY . .

# Install dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Expose port 5000

EXPOSE 5000

# Run app.py when the container launches

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

**Step 4: Build and Run the Flask App in Docker**

1. **Build the Docker image:**
2. docker build -t flask-app .
3. **Run the container:**
4. docker run -d -p 5000:5000 flask-app
5. **Access the application in a web browser:**
   * Open: http://localhost:5000

**Submission Guidelines:**

* Provide screenshots of:
  1. Docker installation (docker --version)
  2. Running Nginx (docker ps -a)
  3. Running Flask application (docker run flask-app output)
* Upload Dockerfile, app.py, and requirements.txt to GitHub and share the repository link.

**Bonus Task (Optional):** Push your Docker image to Docker Hub:

docker login

docker tag flask-app your-dockerhub-username/flask-app

docker push your-dockerhub-username/flask-app