



Ex 07

1300

attach commands and
created images

Exercise 1: Building and Tagging a Docker Image (3*50)

Create a simple Flask application that returns "**Hello from Flask!**" when accessed on port 5000. Build a Docker image for this application, and tag it appropriately.

Tasks:

1. Create a simple Flask application (**app.py**).
2. Write a **Dockerfile** for the application.
3. Build the Docker image and tag it as **flask-app:latest**.

Exercise 2: Pushing and Pulling from Docker Hub (4*50)

Push the Docker image you created in Exercise 1 to Docker Hub, then pull it on a different machine or a virtual machine.

Tasks:

1. **Tag** the Docker image with your Docker Hub username.
2. **Push** the image to Docker Hub.
3. On a different machine, log in to Docker Hub and pull the image.
4. Run the container and verify that it works by accessing the application on port 5000.

Exercise 3: Pausing and Unpausing Docker Containers (3*100)

Understand how to pause and unpause Docker containers by manipulating a running container of your simple Flask application.

Tasks:

1. Run your flask-app image as a container.
2. Pause the container and check the status to confirm it's paused.
3. Unpause the container and check the status to ensure it's running again.

Exercise 4: Saving and Loading Docker Images (5*50)

Practice saving a Docker image to a tar file, transferring it to another directory (simulating a new server), loading it from the tar file, and running it.

Tasks:

1. Use the Docker image from Exercise 1.
2. Save the image to a tar file.
3. Move the tar file to a different directory on the same machine (or transfer to another machine if possible).
4. Load the image from the tar file.
5. Run the container and ensure it operates correctly

Exercise 5: Docker Compose Up and Down (4*100)

Use Docker Compose to manage multi-container applications. Write a docker-compose.yml file for a simple setup with the Flask application and a separate Redis container (even though Flask doesn't interact with Redis in this setup).

Tasks:

1. Write a docker-compose.yml that defines services for both the Flask application and a Redis server.
2. Use Docker Compose to build and run the services.
3. Verify that both containers are running correctly.
4. Use Docker Compose to stop and remove the containers.

THE END