Steps To Pull Docker Images And Push Into Personal Docker Hub

1. Jenkins

Steps to Set Up and Run Jenkins

1. **Pull Jenkins Image**

   docker pull jenkins/jenkins:lts

2. **Tag the Image Before Pushing**

   docker tag jenkins/jenkins:lts anand1827/dev:jenkins

3. **Push Jenkins Image to Docker Hub**

   docker push anand1827/dev:jenkins

4. **Run Jenkins Container**

   docker run --name my-jenkins -d -p 8081:8080 -p 50000:50000 jenkins/jenkins:lts

5. **Verify Running Containers**

   docker ps

6. **Check Logs for Initial Admin Password**

   docker logs fcc7f372c3a0 | grep "initialAdminPassword"

7. **Start Jenkins If Stopped**

   docker start fcc7f372c3a0

   Test in Browser

   Open http://13.235.80.43:8081 in a browser.

---

2. Apache HTTP Server (httpd)

Steps to Set Up and Run Apache HTTP Server

1. **Pull httpd Image**

   docker pull httpd

2. **Tag the Image Before Pushing**

   docker tag httpd anand1827/dev:httpd

3. **Push httpd Image to Docker Hub**

   docker push anand1827/dev:httpd

4. **Run Apache HTTP Server Container**

   docker run -p 80:80 -d httpd

5. **Check Running Containers**

   docker ps

6. **Stop Running Container**

   docker stop ee367d390f6e

7. **Restart httpd**

   docker start ee367d390f6e

   Test in Browser

   Open http://13.235.80.43:80 in a browser.

---

3. Nginx

Steps to Set Up and Run Nginx

1. **Pull Nginx Image**

   docker pull nginx

2. **Tag the Image Before Pushing**

   docker tag nginx anand1827/dev:nginx

3. **Push Nginx Image to Docker Hub**

   docker push anand1827/dev:nginx

4. **Run Nginx Container with Port Mapping**

docker run -itd --name anand -p 40:80 nginx

5. **Verify Running Containers**

   docker ps

6. **Inspect Container**

   docker inspect b7ec3c8bd60b

Test in Browser

Open http://13.235.80.43:40 in a browser.

---

4. MySQL

Steps to Set Up and Run MySQL

1. **Pull MySQL Image**

   docker pull mysql

2. **Tag the Image Before Pushing**

   docker tag mysql anand1827/dev:mysql

3. **Push MySQL Image to Docker Hub**

   docker push anand1827/dev:mysql

4. **Run MySQL Container**

   docker run -p 3306:3306 --name mysql-server -e MYSQL_ROOT_PASSWORD=root -d mysql

5. **Verify Running Containers**

   docker ps

6. **Connect to MySQL Inside Container**

   docker exec -it 3dbe775e1c0b mysql -uroot -proot

---

5. Ubuntu

Steps to Set Up and Run Ubuntu Container with Apache2

1. Pull Ubuntu Image

docker pull ubuntu

3. Tag the Image Before Pushing

docker tag ubuntu:latest anand1827/dev:ubu

4. Push Tagged Image to Docker Hub

docker push anand1827/dev:ubu

5. Run Ubuntu Container

docker run -itd --name my-ubuntu -p 8080:80 ubuntu bash

6. Verify Running Containers

docker ps

7. Access the Container

docker exec -it my-ubuntu bash

8. Update Package Lists

apt update

9. Install Apache2

apt install -y apache2

10. Start Apache Service

service apache2 start

12. Exit the Container

Press Ctrl + P + Q

13. Test in Browser

Open http://13.235.80.43:8080 in a browser.

Explanation for Common Steps Involved in pulling docker images and pushing into personal dockerhub.

1. Pull a Docker Image

Command Syntax:

docker pull <image-name>:<tag>

Explanation:

Downloads a Docker image from a remote registry (e.g., Docker Hub) to your local machine. If no tag is specified, it defaults to latest.

---

2. Tag a Docker Image

Command Syntax:

docker tag <existing-image>:<tag> <your-dockerhub-username>/<new-image-name>:<new-tag>

Explanation:

Assigns a new name and tag to an existing Docker image. This is required before pushing an image to a remote repository.

---

3. Log in to Docker Hub

Command Syntax:

docker login

Explanation:

Authenticates your Docker CLI with Docker Hub by prompting for a username and password.

---

4. Push a Docker Image

Command Syntax:

docker push <your-dockerhub-username>/<image-name>:<tag>

Explanation:

Uploads a tagged Docker image from your local machine to a remote Docker repository.

---

5. Verify Available Images Locally

Command Syntax:

docker images

Explanation:

Lists all Docker images currently stored on your local system, showing their repository, tag, image ID, and size.