Document: Docker Image and Container Creation using EC2 Instance

Step 1: Install Docker on EC2 Instance

For Amazon Linux 2:

- `sudo yum update -y`
- `sudo amazon-linux-extras install docker -y`
- `sudo service docker start`
- `sudo usermod -aG docker ec2-user`

Logout and login again to apply group changes, or run `newgrp docker`

For Ubuntu:

- `sudo apt update`
- `sudo apt install docker.io -y`
- `sudo systemctl start docker`
- `sudo systemctl enable docker`
- `sudo usermod -aG docker ubuntu`

Step 2: Pull Image from Docker Official Website

Docker Hub (https://hub.docker.com) is the official image repository.

Example: Pulling the official Nginx image:

- `docker pull nginx`
- You can replace `nginx` with any official image like `httpd`, `mysql`, `python`, etc.

Step 3: Check Downloaded Images

- `docker images`
- This will list the local Docker images downloaded from Docker Hub.

Step 4: Create and Run a Container from Image

Run an Nginx container:

- `docker run -d -p 8080:80 --name my-nginx nginx`
- `-d` runs container in background (detached mode)
- `-p 8080:80` maps EC2 instance's port 8080 to container's port 80
- `--name my-nginx` names the container

Verify it's running:

- `docker ps`

Step 5: Access the Application

- Go to your EC2 public IP and open:
  `http://<your-ec2-public-ip>:8080`
- You should see the default Nginx welcome page.

Step 6: Manage Containers

- Stop a container:
  `docker stop my-nginx`
- Remove a container:
  `docker rm my-nginx`
- Remove an image:
  `docker rmi nginx`
- Optional: Create Custom Image Using Dockerfile
- Create a new directory:
  `mkdir myapp && cd myapp`
- Create a `Dockerfile`:
  ```
  FROM nginx
  COPY index.html /usr/share/nginx/html
  ```
- Create a simple `index.html`:
  `<h1>Hello from my custom Docker image!</h1>`
- Build the image:
  `docker build -t my-nginx-custom .`
- Run the container:
  `docker run -d -p 8080:80 my-nginx-custom`
- Notes
- Use `docker exec -it <container_id> /bin/bash` to access container shell.
- Always stop containers before removing them.
- Use `docker system prune` to clean up unused resources.