Containerisation

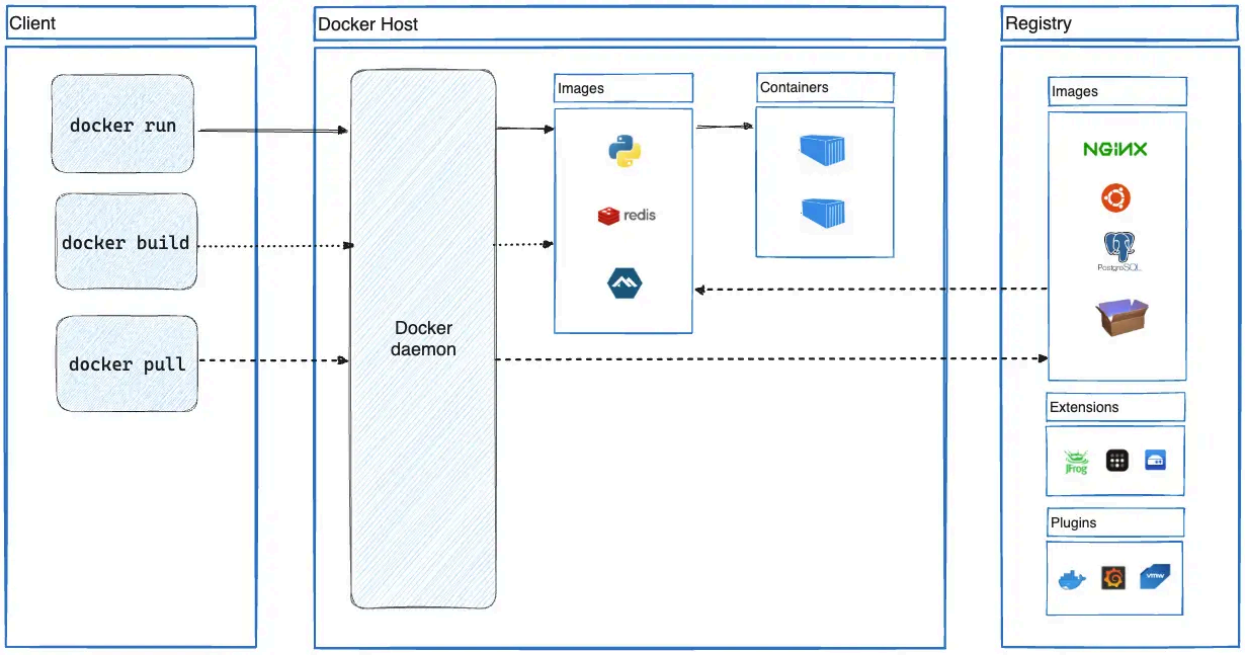
Pre-requisites

- Docker

Activities

Activity 0: Warmup

Answer the questions and share with your group



Questions

1. What is the difference between an image and a container?
2. Where are images stored?
3. What is Docker?
4. Compare the size of an `alpine` and `debian` image, which is smaller?

Activity 1a: Explore Debian and Build on It

Prompts

1. Run a `debian` container and exec into it
2. Use the `ping` command to reach google.com
3. Use the `curl` command to reach google.com
    a. Note the commands used to install `curl`
4. Use `debian:bookworm` as the base image, build a new image (name: `my-nodejs:bookworm`)
    a. Include the `curl` binary. Verify that it exists.
    b. Include `v21.x` of the `node` binary. Verify the binary version.

Hints

- https://www.cyberciti.biz/faq/howto-install-curl-command-on-debian-linux-using-apt-get/
- https://nodejs.org/en/download/package-manager#debian-and-ubuntu-based-linux-distributions
- https://docs.docker.com/build/building/packaging/#building

Activity 1b: Containerize the App

Using the same `Dockerfile` from the last activity, extend on it to containerize our NodeJS application (here).

Prompts

1. Build a new image (name: `express-app:0.1`)
    a. Run the container and verify that your website is reachable

Activity 1c: Environment Variables

Prompts

1. Replace the `msg` variable line to the following,
   ```
   const msg = `Hello from ${ENV} environment`;
   ```
   a. Rebuild the image and verify the changes
2. Replace the `ENV` variable line to the following,
   ```
   const ENV = process.env.APP_ENVIRONMENT || 'undefined';
   ```
   a. Rebuild the image and verify the changes
   b. Provide the environment variable when running the container such that it would not show ‘undefined’

Hints

- https://docs.docker.com/engine/reference/commandline/run/#env

Activity 1d: Git

Prompts

1. Update your README.md to include:
    a. Command to build the image
    b. Command to run the container
    c. Command to curl the website
2. Commit and push to your remote repository

Activity 1e: Build and Run in an EC2 Instance

Prompts

1. Create a t2.micro instance with internet access
2. Install docker
3. Install git
4. Clone your repository
5. Build the image
6. Run the container

Hints

- https://www.cyberciti.biz/faq/how-to-install-docker-on-amazon-linux-2/

(Bonus) Activity 2: Containerize a Docker App

Prompts

1. Read the README.md in the repository
    a. Repo: `https://github.com/seanlim1/sctp-ce-python-demo`
    b. Branch: `before-containerizing`
2. Create the `Dockerfile`
3. Build a new image (name: `flask-app:0.1`)
    a. Run the container and verify that your website is reachable