

What I learned about docker

Bedirhan Sen

2024-09

1 Introduction

Docker is a fix for problems that program works in one machine does not work for another. This is maybe related to missing files, software versions, different config settings. For now I understood it like an advanced version requirements.txt files and virtual env for running python programs . With Docker we package an application and run it any machine.

2 Virtual Machines and Containers

VM: an abstraction of machine(physical hardware) hypervisor: tools to build and manage vm's like WMware. problems with vm is that each application with unique settings and dependencies needs different vm and each vm needs a full OS to run. Therefore they are slow to start and resource intensive. They allocate RAM, CPU and disk space.

container: an isolated environment running an application, they are lightweight, they allow running multiple apps in isolation, they use the OS of the host therefore they are faster. They also need fewer hardware resources.

3 Docker Architecture

Client-Server(Docker Engine) relationship. Container uses the kernel of the host.

4 Development workflow

Add a dockerfile to the application. This file is used to package up this application into an docker image. This image contains, a cut-down os, a runtime environment, application files, third-party libs. Example Dockerfile

```
# Use the official Python image as a base
FROM python:3.9-slim-buster

# Set environment variables
ENV PYTHONUNBUFFERED 1
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0

# Set the working directory in the container.
# all the instructions after this will assume we are in this directory
WORKDIR /app

# Copy the requirements file and install dependencies
```

```
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

# Copy the application code into the container
COPY . /app

# Expose port 5000 to the outside world
EXPOSE 5000

# Run the Flask app when the container launches
CMD ["flask", "run"]
```

after this run

```
docker build -t {identifier_tag} {Dockerfile dir} #builds image
docker image ls # to see images in the host computer
docker run {identifier_tag} # run the image
```

5 Use ROS in a docker container

```
docker pull ros
docker run -it ros
```

Especially important for using different versions of ros in your machine at the same time.