

Docker and Containers

IEEE Spring 2024



We'll be Using the Raspberry Pi's Tonight

- Installing Docker on non-Linux systems is a PITA.
- Technically runs in a virtual machine on Windows and MacOS.
 - Will be explained in a bit...
- For future reference, you can install on your platform of choice with these instructions: <https://docs.docker.com/engine/install/>

Virtual Machines

- Simulate a complete physical machine with it's own CPU, GPU, memory, storage, and operating system.
- Can run pretty much any operating system, regardless of the host operating system.
- Resources are allocated up front and dedicated to a particular VM, making scaling harder.

Containers

- Share the host operating system's kernel and system calls.
- Contain an application and its dependencies, but don't have dedicated resources.
- Ephemeral unless otherwise specified.
- Reproducible.
- Generally Linux native because of the use of specific kernel features.
 - Docker specifically claims Windows native compatibility
 - <https://docs.docker.com/engine/faq/>

Why Use Containers?

- They consume much less storage space.
 - 1GB container * 1000 containers = just over 1GB of storage used.
- They are much faster to boot.
 - VM's can take minutes to get up, containers are ready almost immediately.
- They are easier to scale.
 - Host resources are just a pool that containers dip into.
- Reproducibility.
 - No longer have to say "it works on my machine".
 - Just ship your machine!

An Important Distinction

- Docker != Containers
- There is an open source specification for containers.
 - <https://opencontainers.org/>
- Docker is an engine for running containers.
- Podman is another such engine for running containers.

Podman vs Docker

| | Docker | Podman |
|--------------|---------------------------------|---|
| Architecture | Client-server (daemon) | Daemonless |
| Security | Optional rootless mode | Rootless by default |
| Ease of use | Easier for beginners | A bit more alien |
| Features | Orchestration and image sharing | Focus on running containers, integrates with Kubernetes |

Time For Hello World

```
docker run hello-world
```


Something A Bit More Interesting

```
docker run <image> <command>
```

```
docker run --interactive --tty ubuntu bash
```

```
apt update
```

```
apt install neofetch
```

```
neofetch
```

Deterministic Builds With A Dockerfile

```
FROM ubuntu:22.04
```

```
RUN apt-get update
```

```
RUN apt-get upgrade -y
```

```
RUN apt-get install -y neofetch
```

^^^ put these lines in a file named Dockerfile

```
docker build -f Dockerfile -t neofetch .
```

```
docker run -it neofetch
```

Docker Compose

- Simplifies controlling multiple containers at once.
- Single command to bring a number of services online.
- Deals with services relying on one another.
- Configured in a single YAML file.
- YAML requires spaces and not tabs, apologies in advance for the

Docker Compose WordPress Example

<https://gist.github.com/dkrautha/c6f45c6840df9e4a2e7fea83d2340d50>

Save the contents in this gist into a file named docker-compose.yml

In the same folder as the file you created, run the following

```
docker compose up
```

Give it about a minute for everything to setup, and then head to

<https://localhost:8675>

Further Resources

- <https://docs.docker.com/compose/>
- https://docs.docker.com/engine/reference/commandline/compose_up/
- <https://docs.docker.com/engine/reference/builder/>
- <https://podman.io/>
- <https://hub.docker.com/>
- <https://quay.io/>