

Dockers!

By Thomas and Faisal



SIGPWHY

What is Docker?

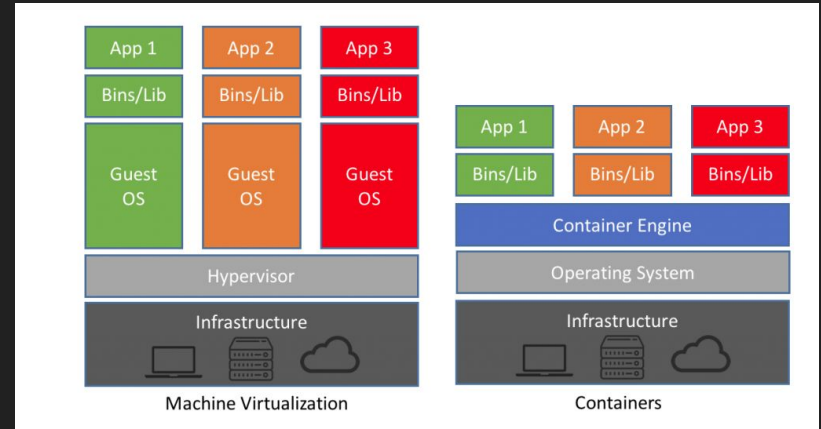
- A tool that allows you to run and create applications in isolated environments (containers).
- The containers each include their own namespace, libraries, and system tools.
- Can run in the background as long as your computer is on!



SIGPWNY

Docker vs. Virtual Machines

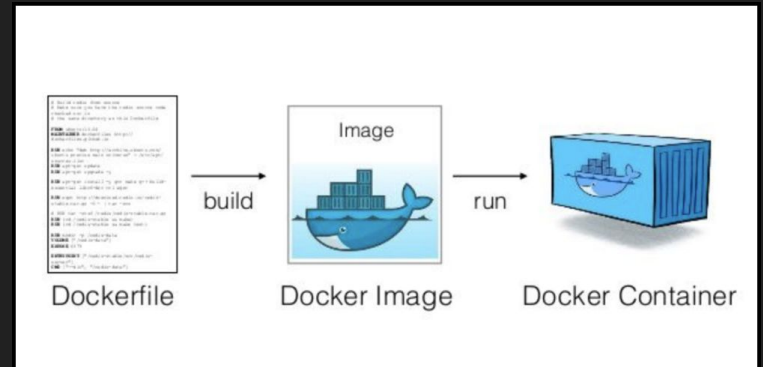
- Docker containers are significantly more lightweight than virtual machines.
- Dockers share the operating system with the host whereas virtual machines each have their own operating system.
 - Containers do 'emulate' an operating system using files though
- Aims to solve the 'doesn't run on my computer' problem.



SIGPWNY

Dockerfiles, Images, Containers - What's the difference?

- Dockerfiles are simple text files that contain commands for Docker to 'build' us an image. The commands can specify the type of operating system and version to build and what files to store in our environment.
- Images file that contains the source code, libraries, dependencies, tools, and other files needed for an application to run.
- Containers are the isolated environment where you can run your code and/or applications!



SIGPWNY

Dockerfiles

- Basic text files!
- They just contain a list of commands that Docker will run to ‘build’ and image file.
- Let’s run a quick demo!



SIGPWNY

Pull docker Images

- Docker Hub actually contains a [bunch of official image files!](#)
- Woohoo! That means we don't have to build our own dockerfiles everytime.
- Let's run another quick demo!



SIGPWNY

Important Docker Commands

- `docker build -t dockername .`
 - Uses the dockerfile to build an image with the name: dockername
- `docker run --rm -it dockername`
 - Runs the image file.
 - `--rm` 'closes' the container when you exit
 - `-it` makes it interactive (so you can use commands!)
- `docker exec -it dockername`
 - Runs a command inside a container
- `docker container/image ls`
 - Lists of all the docker images we can run
- `docker cp`



SIGPWNY