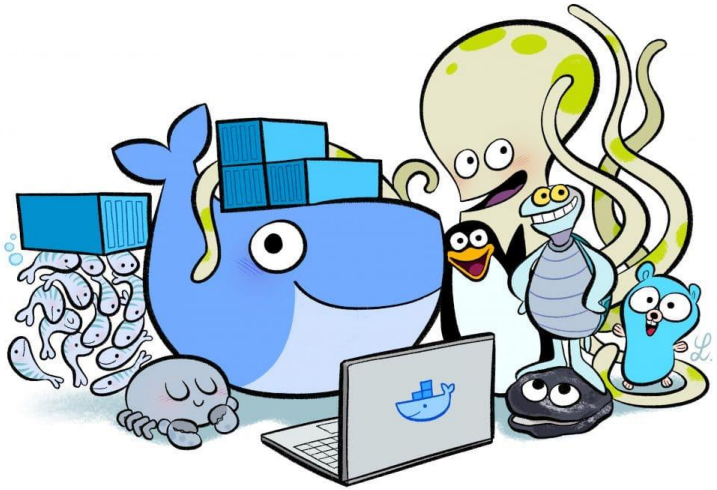
CSCI 5409: Adv. Topic In Cloud Computing

TA: Purvesh Rathod



WHALE HELLO THERE!

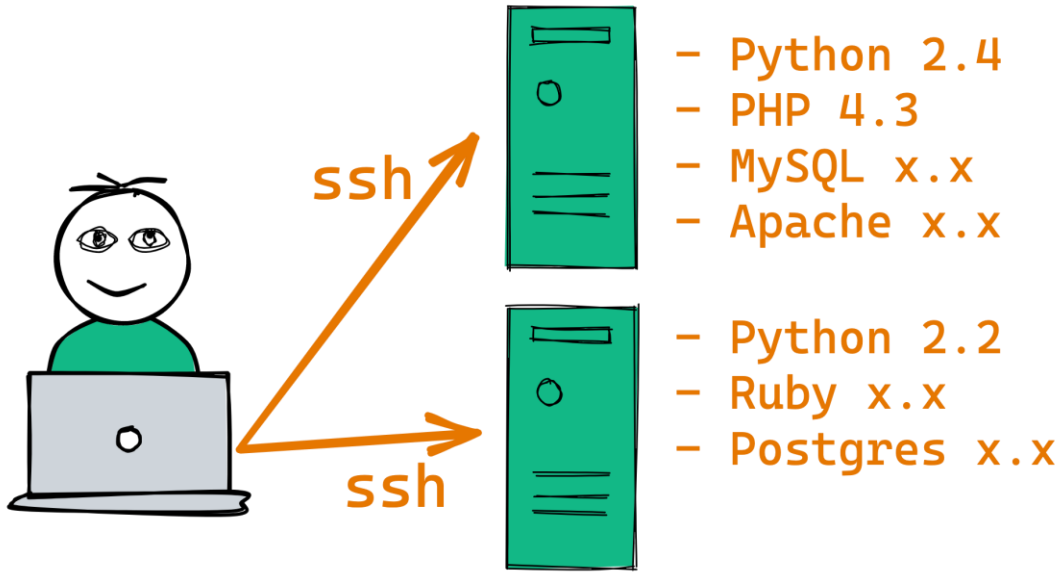
Tutorial 1: Docker Introduction

Tutorial Summary

- Learning outcomes
  - Deployment techniques before Docker
  - Introduction about Docker
  - Docker installation
  - Docker hands-on

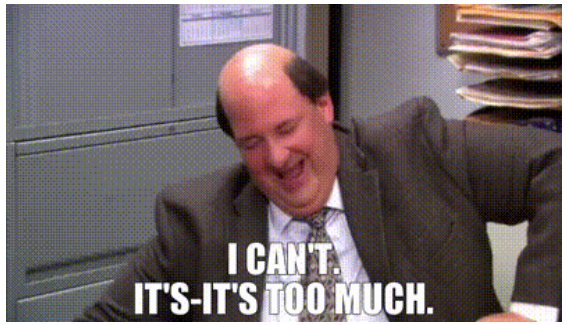
Before Docker Age (B.D. Age)

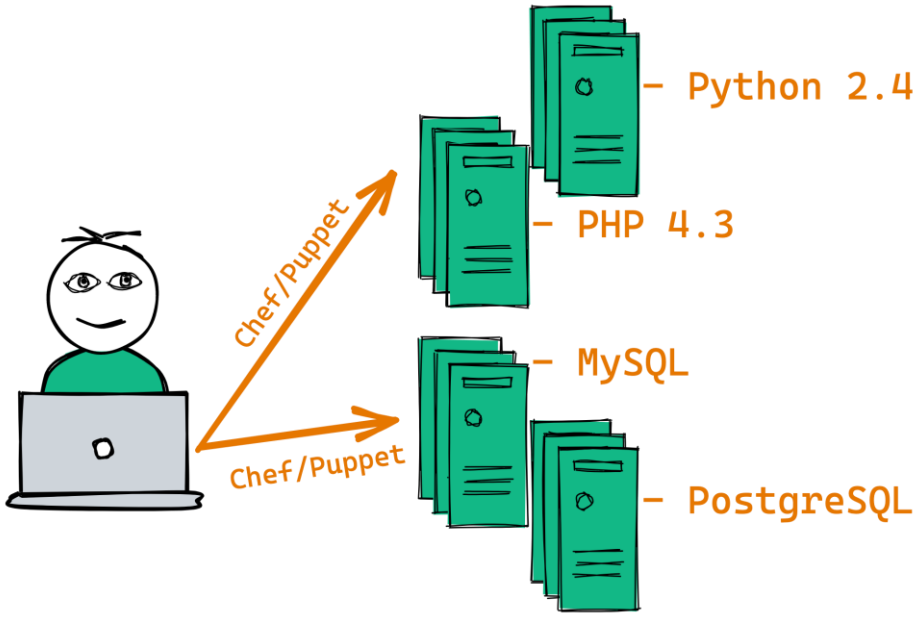
- To deploy an application the DevOps Engineer or system admin would have to perform tedious steps such as:
  - Setting up the server environment.
  - Installing the configurations and required libraries for the application.
  - Deploying the code to the servers Dev ➞ Production.
  - Running/managing the applications.



Before Docker Age (B.D. Age)

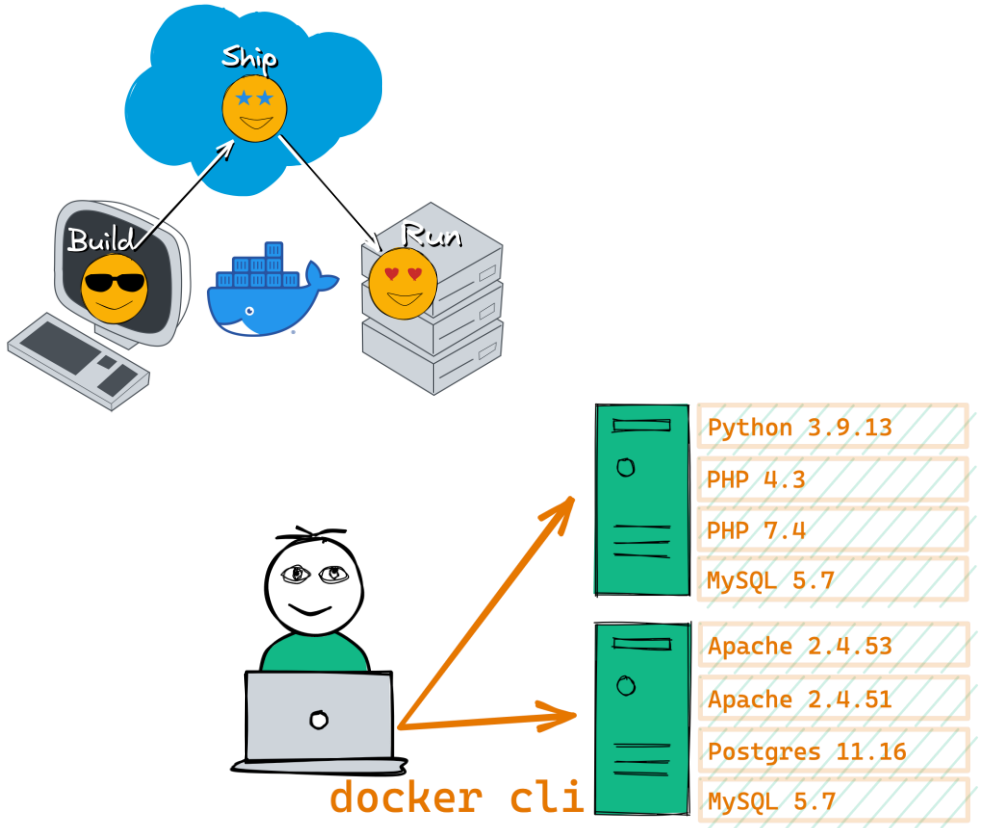
- Virtual Machines to rise! 🔥
- Abstract out the server machine and start running applications on the small VMs.
- E.g. Chef/Puppet to manage the growing number of the servers.
- With the increase in VMs it would make it hard for the admins to manage all the servers at once event with the cloud.





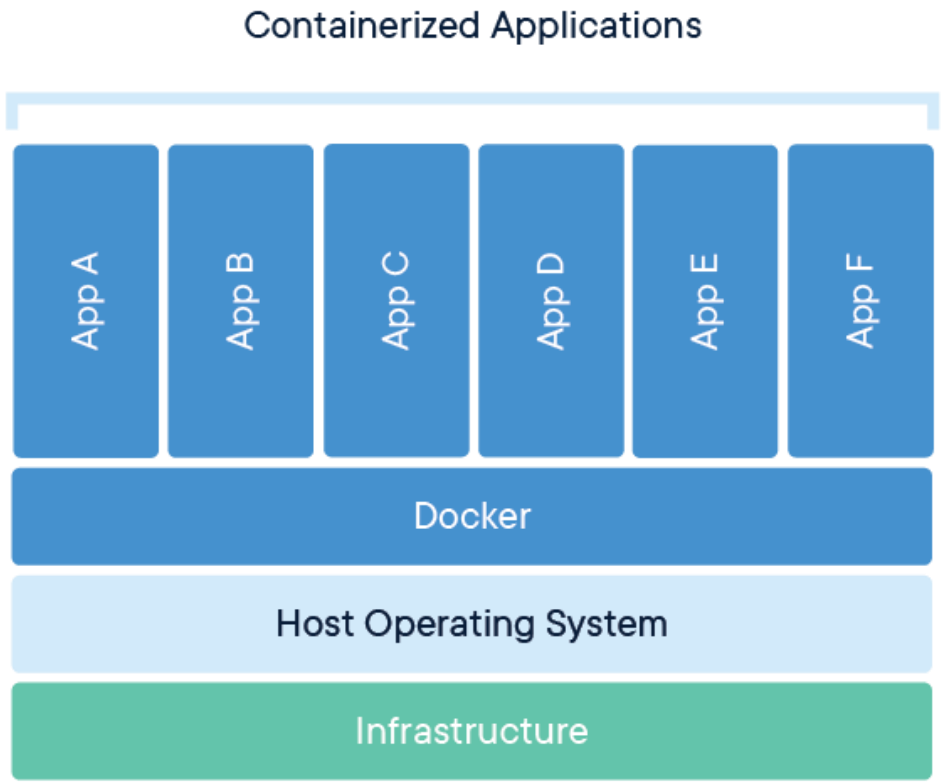
Docker!

- Docker is an open platform that enables developers to build, run, deploy, update and manage containers.
- A **containers** is a standard unit of software that packages application source code, executable components, with the operating system (OS) libraries and dependencies required to run that code in any environment.
- Docker was not the inventor of the containers, but it made them easier to use.



Containerization

- Containers are packages that have everything an application needs to run in any environment.
- Docker uses two Linux Kernel features, called **namespaces** and **cgroups** (control groups), to isolate your app so it can't see the rest of the host by default. To the app, the only files it sees are the ones in the container image. It sees no other processes outside the container.
- A container gives isolation like a **VM***, where they get their own IP address, file system and their own process space but **you don't need a whole OS like a VM** to run your application. This makes container very small and lightweight.



More about containers…

- **Security:** What you put in the container is immutable. Event the software inside the container cannot modify the container. Changing things will produce a new container.
  - By default, everything is turned off: all the ports are closed.
  - Ability to control resource usage (limit memory, read-only file system)
- **Lightweight:** Containers store only what's needed to run the software they contain, **nothing else**.
- **Elimination of configuration errors:** Once configured, the container runs the same, everywhere. What runs in your machine will be running the same in different environments.

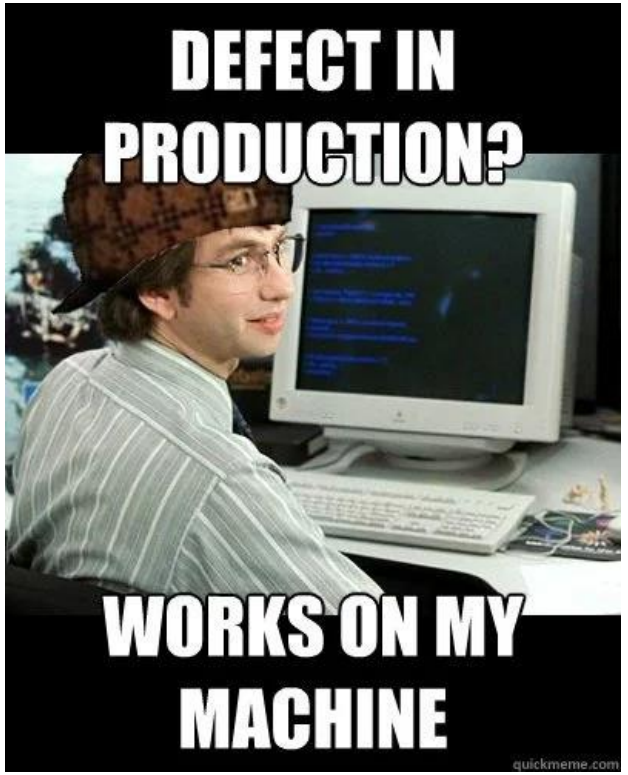
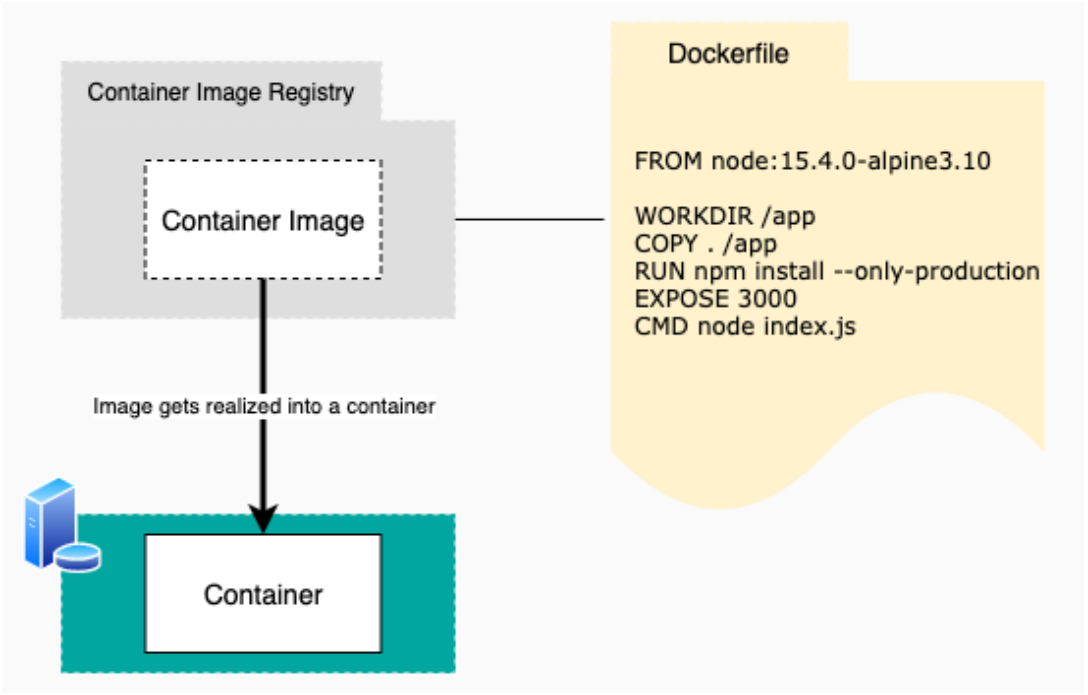


Image vs. Container

- An Image is the application we want to run. It is a read-only immutable template that defines how a container will be realized.
- A Container is a runtime instance of that image running as a process.
- Docker image templates can exist in isolation, but containers can't exist without images.
- You can have many containers running off the same image.



Let's Recap

MAJOR INFRASTRUCTURE SHIFTS

Mainframe to PC
90'S

Baremetal to Virtual
00'S

Datacenter to Cloud
10'S

Host to Container (inc. serverless)

Getting Started with Docker & Docker Hub

- **Docker Hub** is a service provided by Docker for finding and sharing container images.
- Install docker on your machine: Install Docker Engine | Docker Documentation
- Getting started with Docker: A Docker Tutorial for Beginners (docker-curriculum.com)
- Play with Docker without need to install (by Docker Inc.) Play with Docker (play-with-docker.com)

Docker Commands

- **Show docker version:** docker version
- **Help command:** docker help
- **Running nginx server locally**: docker run --detach –p 80:80 nginx
- **Show images:** docker images
- **Show running containers:** docker ps
- **Container logs:** docker logs <container-id>
- **Going inside the container:** docker exec -it <container-id> bash
- **Container stats (Memory/Disk):** docker container stats
- **Stop container:** docker container stop <container-id>
- **Remove container:** docker container rm [-f] <container-id>
- **Remove image:** docker image rm <image-id>

Questions?

References

[1] M. Dock, “Docker: Accelerated, containerized application development,” *Docker*, 10-May-2022. [Online]. Available: https://www.docker.com/. [Accessed: 08-May-2023].

[2] “Docker overview,” *Docker Documentation*, 08-May-2023. [Online]. Available: https://docs.docker.com/get-started/overview/. [Accessed: 08-May-2023].

[3] “What is Docker?,” *Ibm.com*. [Online]. Available: https://www.ibm.com/topics/docker. [Accessed: 08-May-2023].

[4] The TechCave, “What is Containerization? What is Docker? | Containerization vs Virtualization,” 02-Jul-2020. [Online]. Available: https://www.youtube.com/watch?v=yLNMMdyak6k. [Accessed: 08-May-2023].

[5] B. Fisher, *what-is-docker.md at main · BretFisher/udemy-docker-mastery*. .

Thank You!