Introduction to Docker and Singularity

Vivek Venkatraman Krishnan

Fundi tutorials, March 2023

Outline

- What is a container?
- Why a container?
- Containers vs VMs
- Introduction to Docker
- Docker → Singularity
- Very short introduction to Singularity
- Using Singularity on our machines
- Using nextflow with Singularity

What is a container?

- A software that packages up other software and dependencies

What is a container?

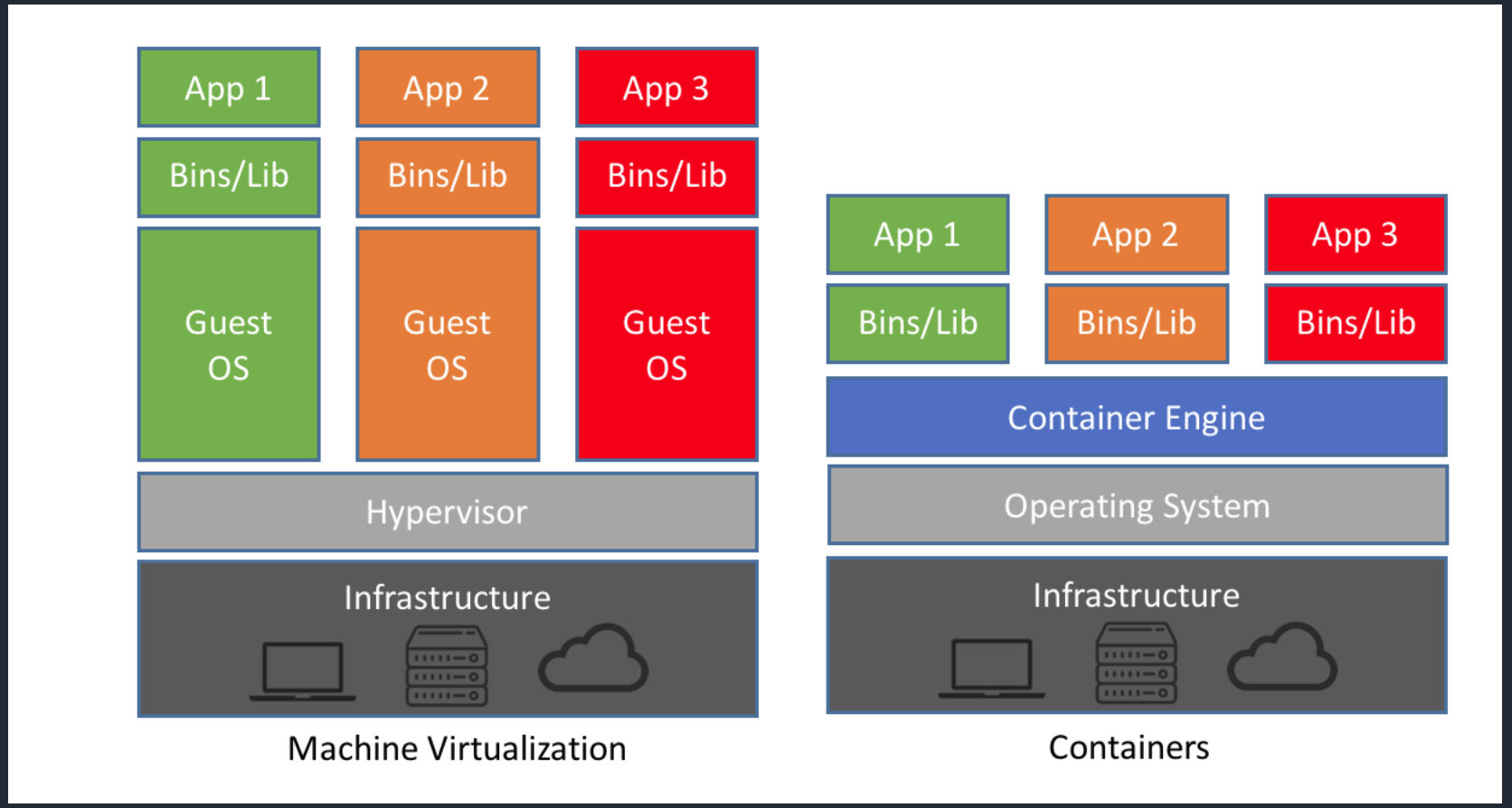
- A software that packages up other software and dependencies

Software X
Version Y

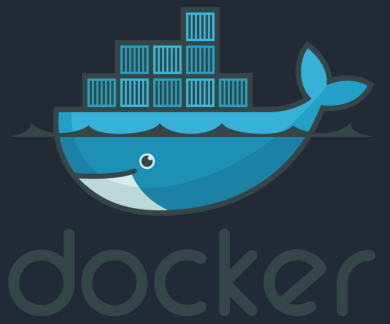
Software X
Version Z

Why use containers?

Containers vs Virtual Machines

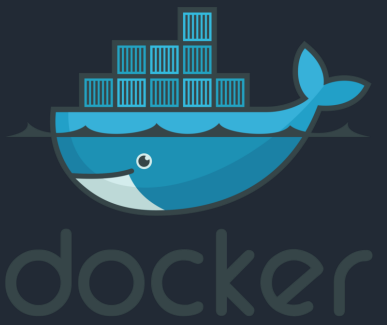


Introduction to docker



- A platform for developing, shipping and running containerised applications
- Not the first, not the last but the most popular
- Used mostly for “microservices”, hacked to work for Science / HPC applications

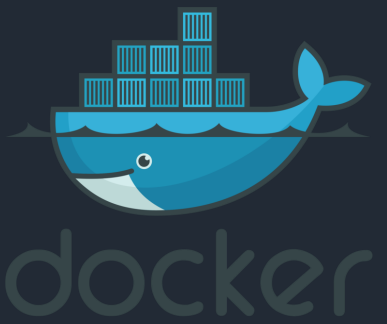
All docker jargons



- Docker container: An container for software. Like an Amazon delivery box!
- Docker image: A blue print to build such container: Your Amazon order!
- Docker repository: Save different versions of your containers.
- Docker registry: A repository of repositories.

  Eg: Docker Hub

Let’s try!



- Run a prebuilt docker

  docker run -it ubuntu:latest

- How did it run this? Let’s try it separately

  docker pull ubuntu:bionic

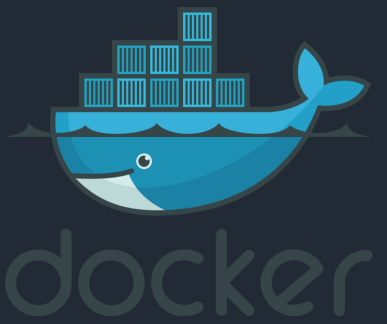
  docker run –it ubuntu:bionic

- How did it know where to download from?

  docker pull docker://ubuntu:bionic

  docker run –it ubuntu:bionic

Docker Hub



https://hub.docker.com/

- Default docker “registry”
- Both public and private images
- Can be used to host / transfer / extend images

Use your own Docker container

- Make a dockerfile
- Build a docker image
- Spawn a docker container

Docker directives

- Start with a base container: FROM
- Install everything you want: RUN
- Set environments as needed: ENV
- Change directories: WORKDIR
- Copy data/code into container: COPY
- Set an entry point: ENTRYPOINT
- Run a command when starting: CMD
- Identify the dockerfile with a name: LABEL

Docker manipulation

- docker build –t <name> .
- docker images
- docker inspect <image_name>
- docker run –it <name>
- docker start <name>
- docker stop <name>
- docker attach <name>
- docker rm <containter>
- docker rmi <image>

Docker manipulation

- Run docker as a daemon: -d
- Expose and use ports: -p and -expose
- Use host directories inside the container: -v
- docker save/ load
- docker tag
- docker kill

Singularity

- Academic solution to running containers on HPC
- Native support for HPC hardware like infiniband, GPUs
- Native support for HPC filesystems: No layers, just a flat file
- Native support for non-root file access. Outside user = inside user

How to create a singularity container?

- Build from a definition file:
  - sudo singularity build test.def
  - Spython library can convert docker file to singularity definition
- Easier to do:
  - Convert docker image to singularity image
  - Official docker image available to do this: GitHub - singularityhub/docker2singularity.
  - docker run -v /aux/pc137a/dockers/docker_tutorial:/output -v /aux/pc137a/dockers:/tmp -v /var/run/docker.sock:/var/run/docker.sock --rm quay.io/singularity/docker2singularity:v3.7.0 py311

How to run a singularity container?

- singularity exec –B /host/dir:/container/dir <image>
- singularity shell –B /host/dir:/container/dir <image>
- SINGULARITY_BIND


APPTAINER