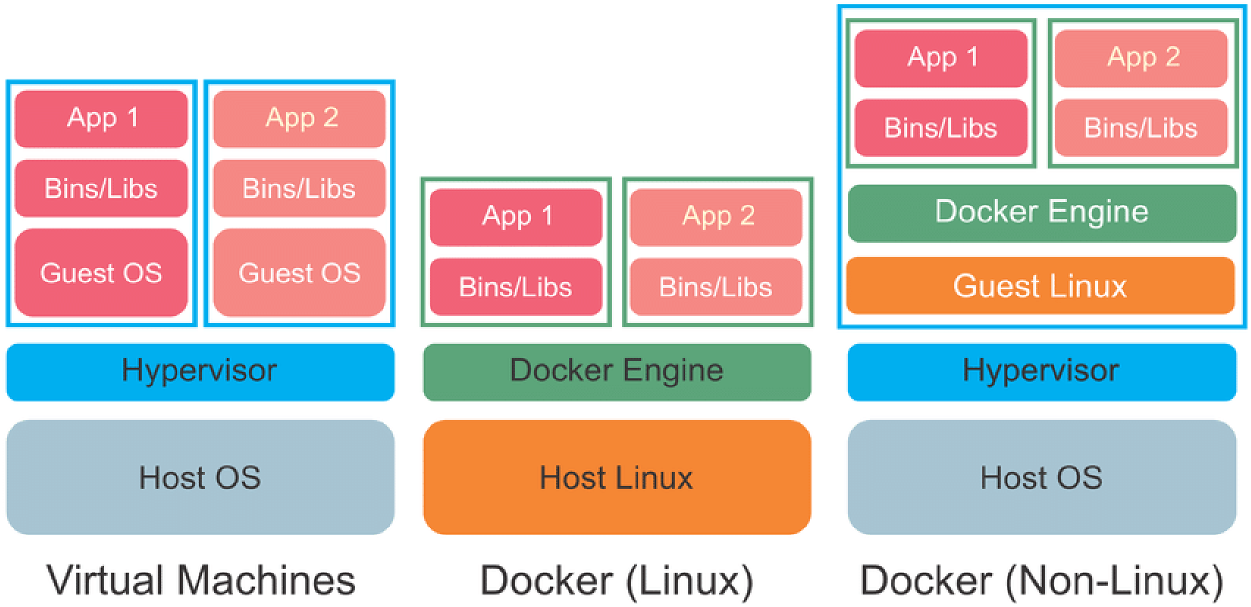
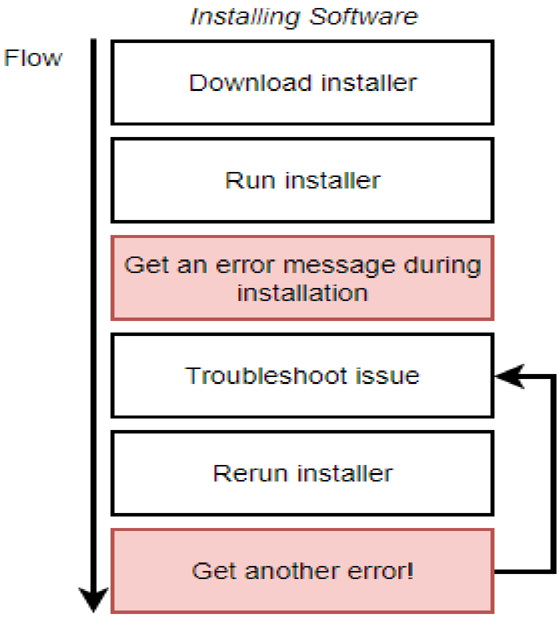
Dockers

VM v/s container



Why Docker?

1. To install and run software easily



This is what Docker is trying to fix

What is docker?

Docker is an ecosystem around creating and running containers.

Docker ecosystem

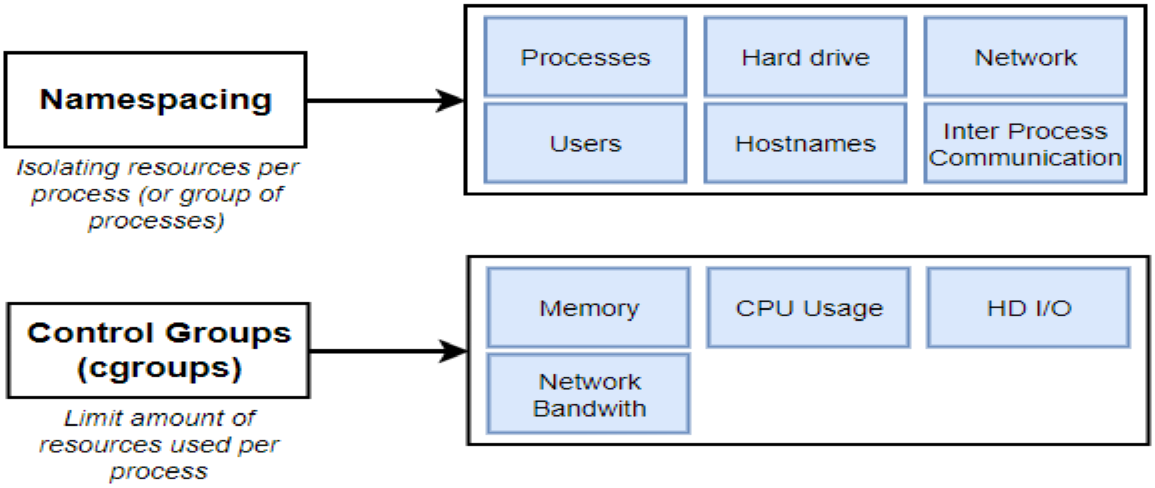
- Docker clients
- Docker server
- Docker hub
- Docker images
- Docker machine
- Docker compose

Underlying technology

- Docker is written in the Go programming language
- Takes advantage of several features of the Linux kernel to deliver its functionality.
- Docker **uses a technology called namespaces to provide the isolated workspace called the *container*.**
- When you run a container, Docker creates a set of *namespaces* for that container.

Namespacing and Control Groups:

- **This is specific to linux.**
- These namespaces **provide a layer of isolation.**
- Each aspect of a container runs in a separate namespace and its access is limited to that namespace.

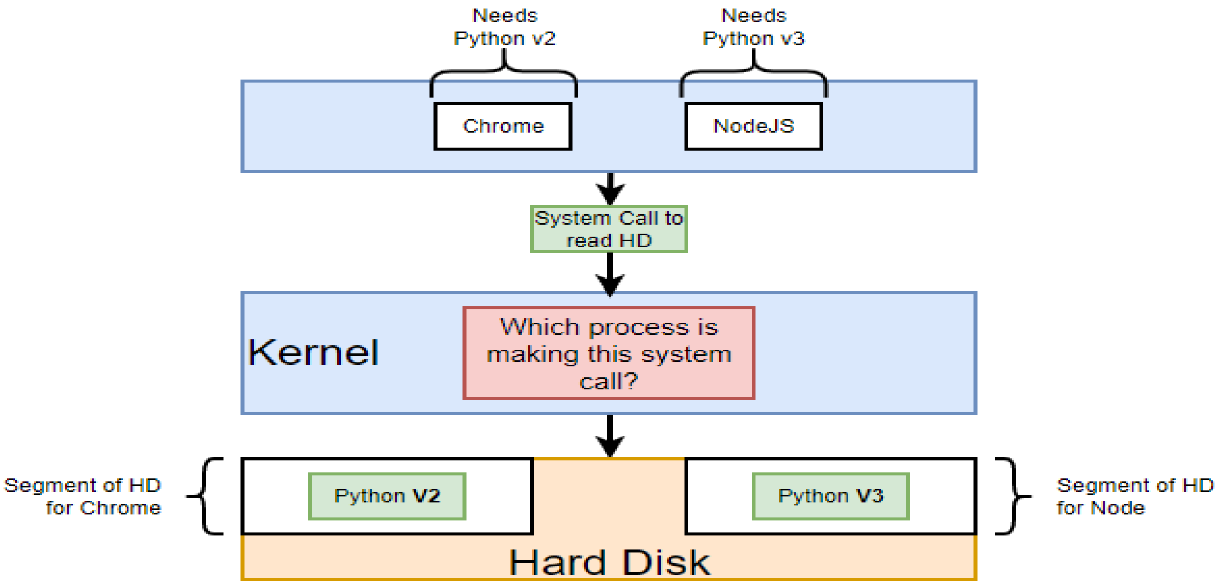


Namespacing
Isolating resources per process (or group of processes)
Processes
Hard drive
Network
Users
Hostnames
Inter Process Communication
Control Groups (cgroups)
Limit amount of resources used per process
Memory
CPU Usage
HD I/O
Network Bandwith

Background of Operating system working

OS : interface between software and hardware.

For example (just a analogy)



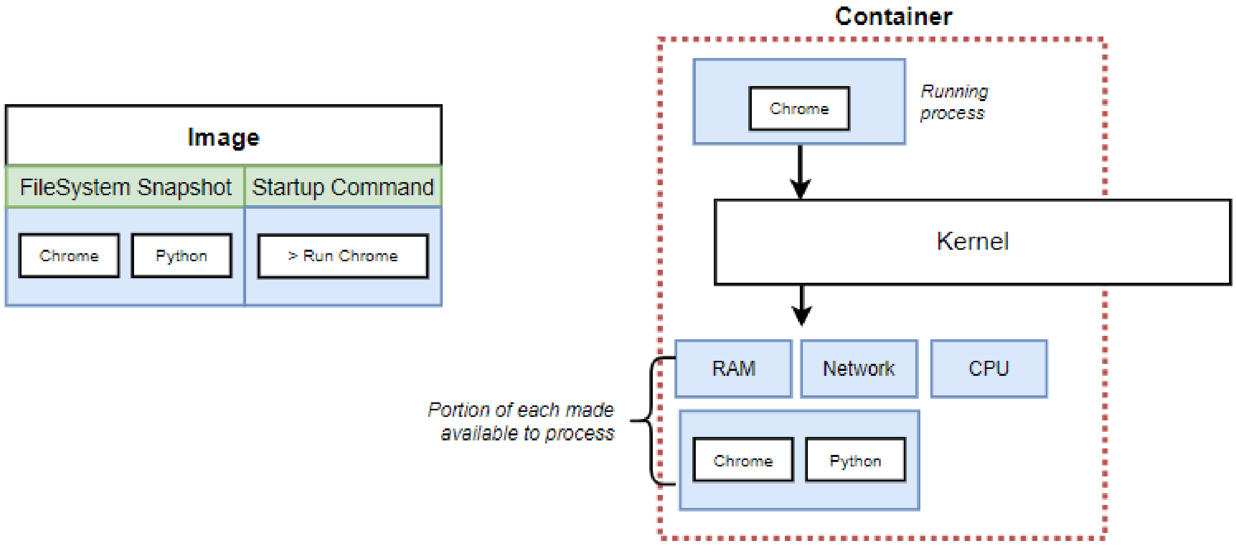
What is an Image?

- Single file with all the dependency and configuration required to run the application.
- Any time that we talk about an image, we're really talking about a **file system snapshot.**

What is a Container?

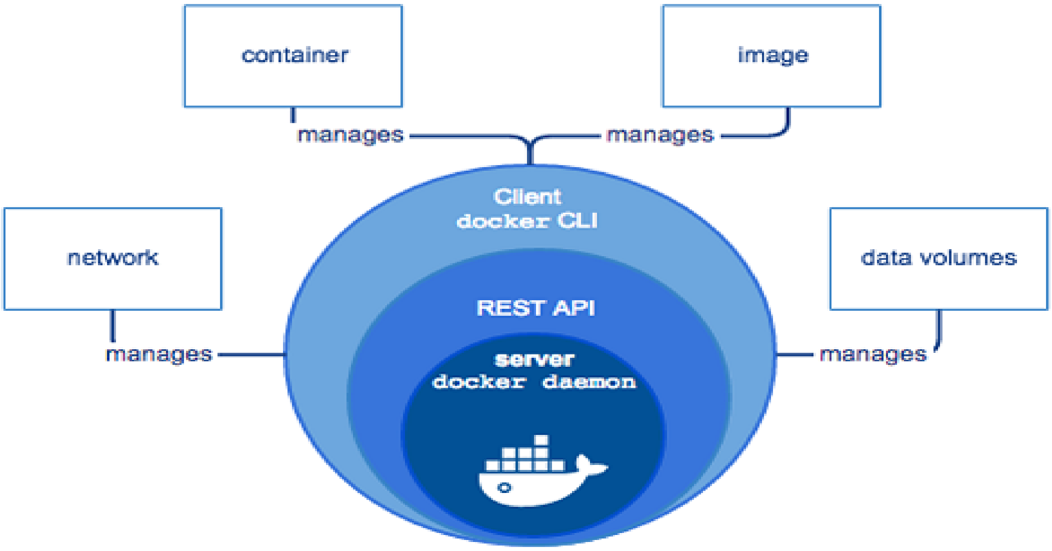
- Running instance of image.
- Docker container, you really **should not think** of these as being like a **physical construct** that exists inside of your computer.
- **Instead, a container is really a process or a set of processes that have a grouping of resources specifically assigned to it.**

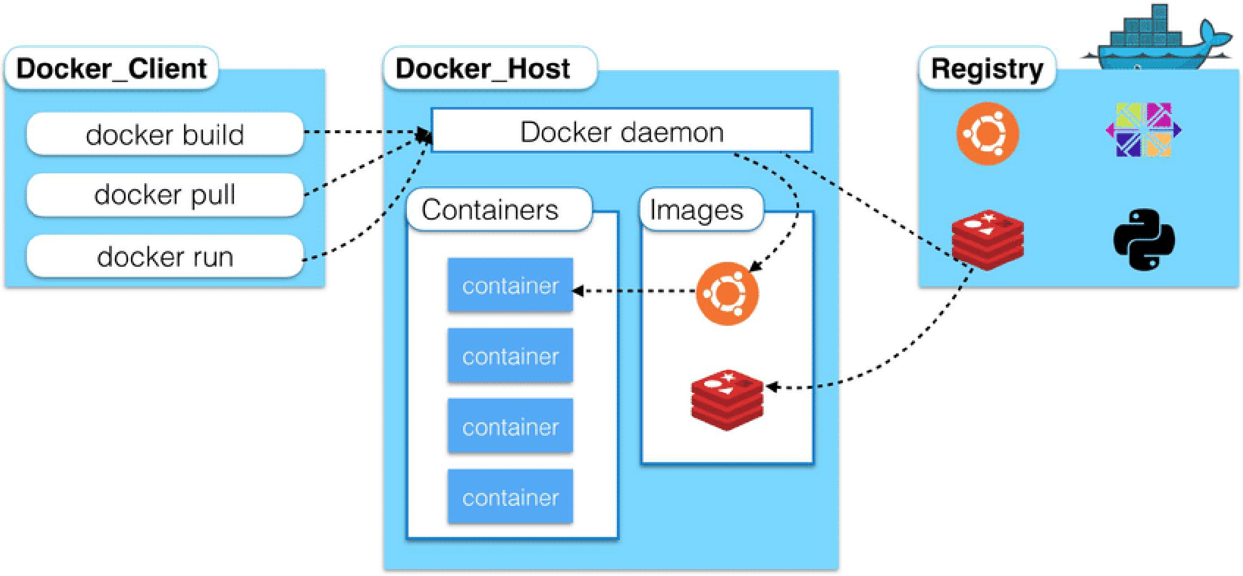
Relationship between image and container



- Any time that we talk about an image, we're really talking about a file system snapshot.
- So this is essentially kind of like a copy paste of a very specific set of directories or files.
- And so we might have an image that contains just Chrome and Python.
- An image will also contain a specific startup command.
- So here's what happens behind the scenes when we take an image and turn it into a container.
- First off, the kernel is going to isolate a little section of the harddrive and make it available to just this container.
- And so we can kind of imagine that after that little subset is created, the file snapshot inside the image is taken and placed into that little segment of the hard drive.
- And so now inside of this very specific grouping of resources, we've got a little section of the hard drive that has just chrome and python installed and essentially nothing else.
- The start up command is then executed, which we can kind of imagine in this case is like start up chrome, just run chrome for me.
- And so chrome is invoked.
- We create a new instance of that process and that created process is then isolated to this set of resources inside the container.

Docker Architecture





- Docker uses a **client-server architecture**.
- The Docker **client and daemon *can* run on the same system,** or you can connect a Docker client to a remote Docker daemon.
- The **Docker client and daemon communicate using a REST API**, over UNIX sockets or a network interface.

Docker client

- The **Docker client (docker)** is the primary way that many Docker users interact with Docker. When you use commands such as docker run, the client sends these commands to dockerd, which carries them out.
- The docker command uses the Docker API.
- The Docker client can communicate with more than one daemon.

Docker daemon

- The **Docker daemon (dockerd)** listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes.
- A daemon can also communicate with other daemons to manage Docker services.

Docker registries

- A **Docker *registry* stores Docker images.**
- Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.
- When you use the **docker pull or docker run commands, the required images are pulled from your configured registry.**
- When you use the **docker push command, your image is pushed to your configured registry.**

Docker Desktop

- Docker Desktop is an easy-to-install application for your Mac, Windows or Linux environment that enables you to build and share containerized applications and microservices.
- Docker Desktop includes
  - Docker daemon (dockerd),
  - Docker client (docker),
  - Docker Compose
  - Docker Content Trust
  - Kubernetes

    - Credential Helper.

Installation

- Refer : https://docs.docker.com/desktop/

Reference

- https://docs.docker.com/get-started/overview/#:~:text=Docker%20uses%20a%20client%2Dserver,to%20a%20remote%20Docker%20daemon.