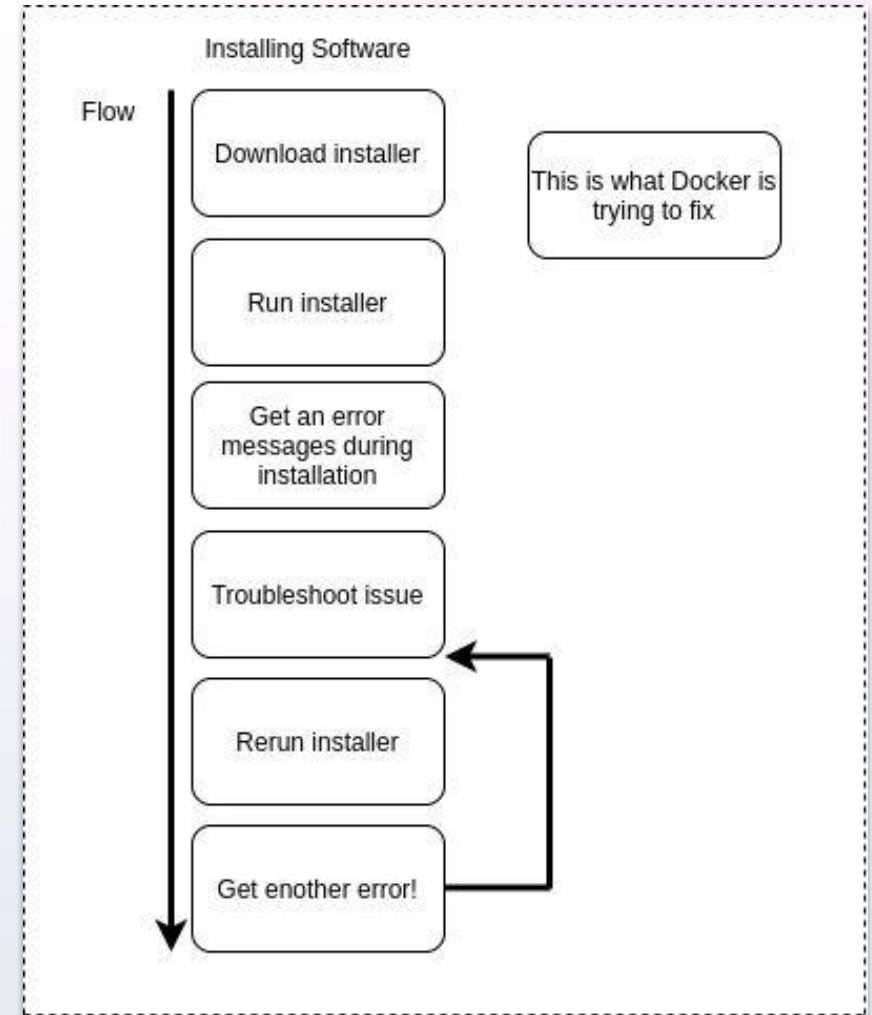
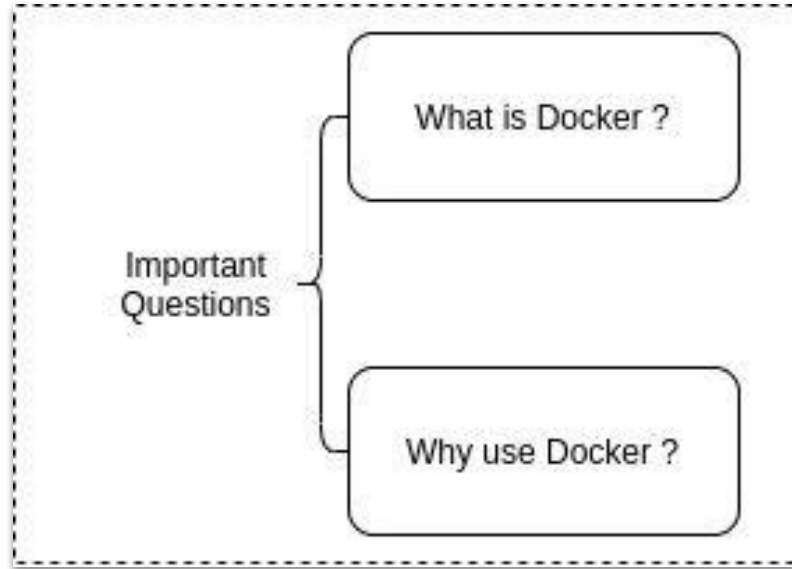


Docker Basics & Building Custom Images

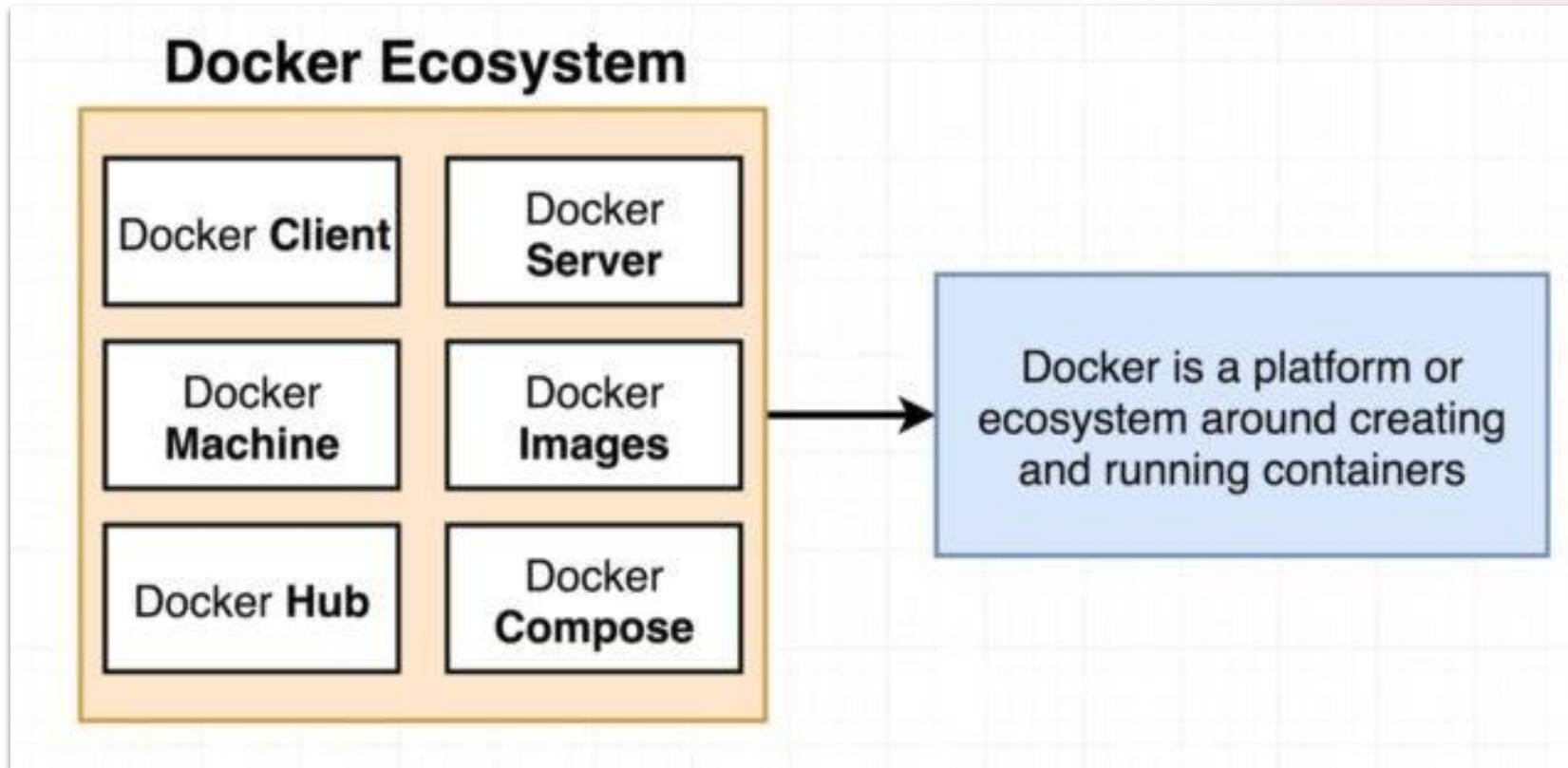
Mocktar ISSA
Full-Stack Software Engineer ,



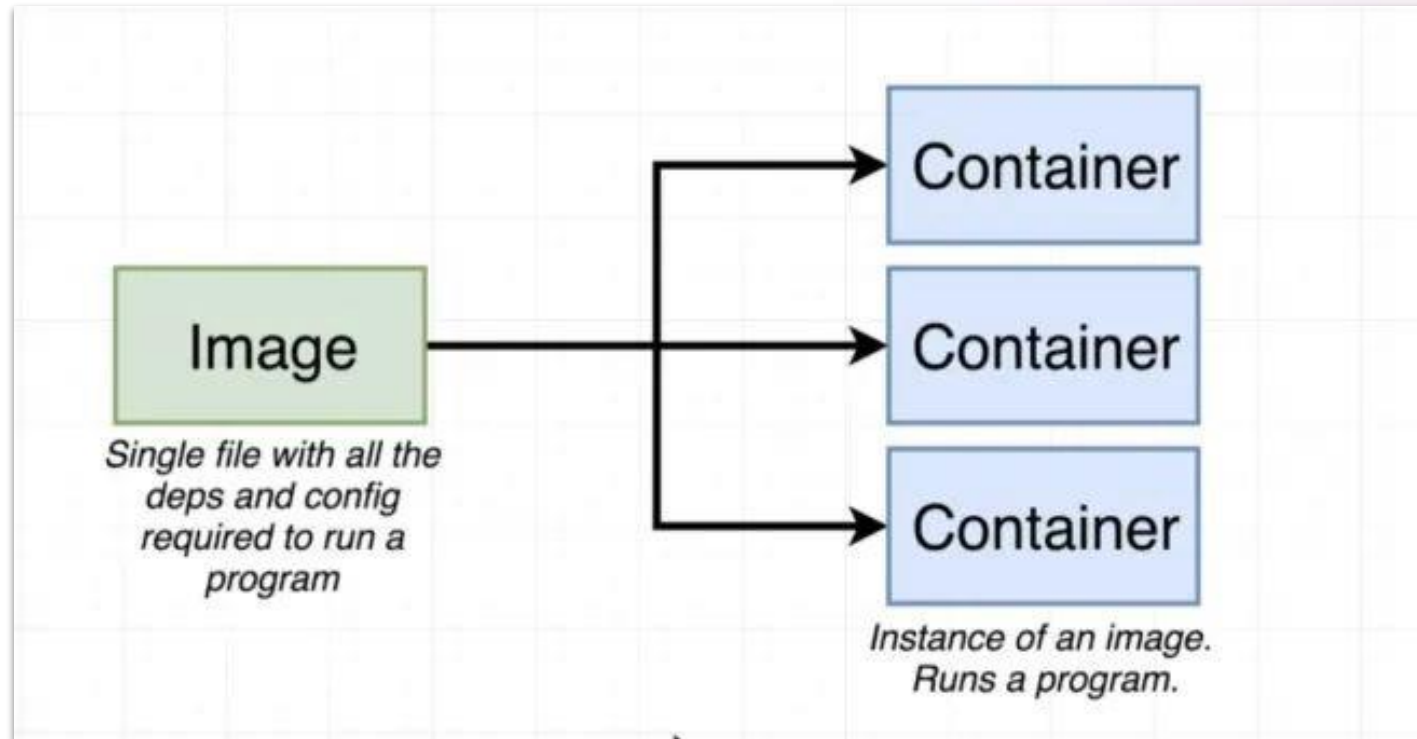
Why use Docker ?



What is Docker ?



What is Docker ?

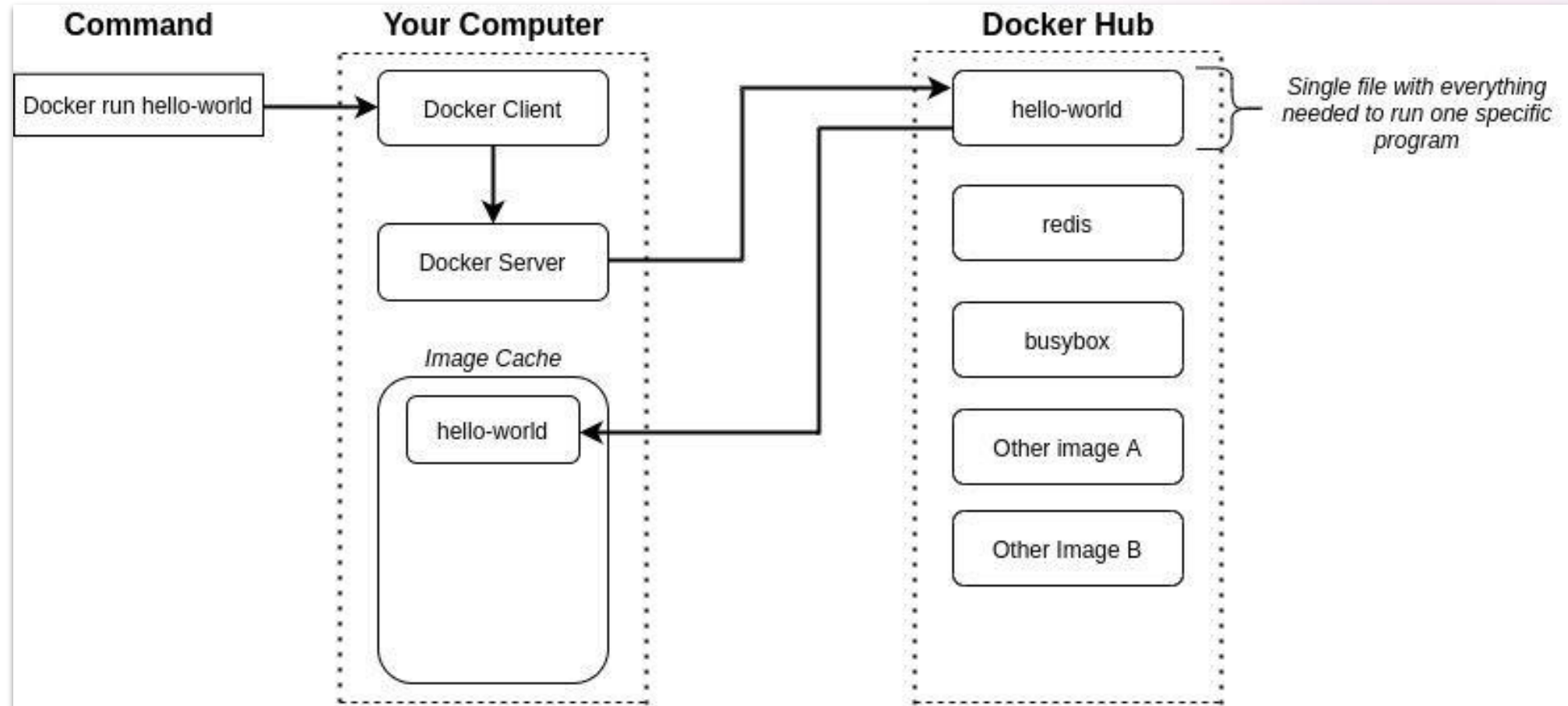


Docker install

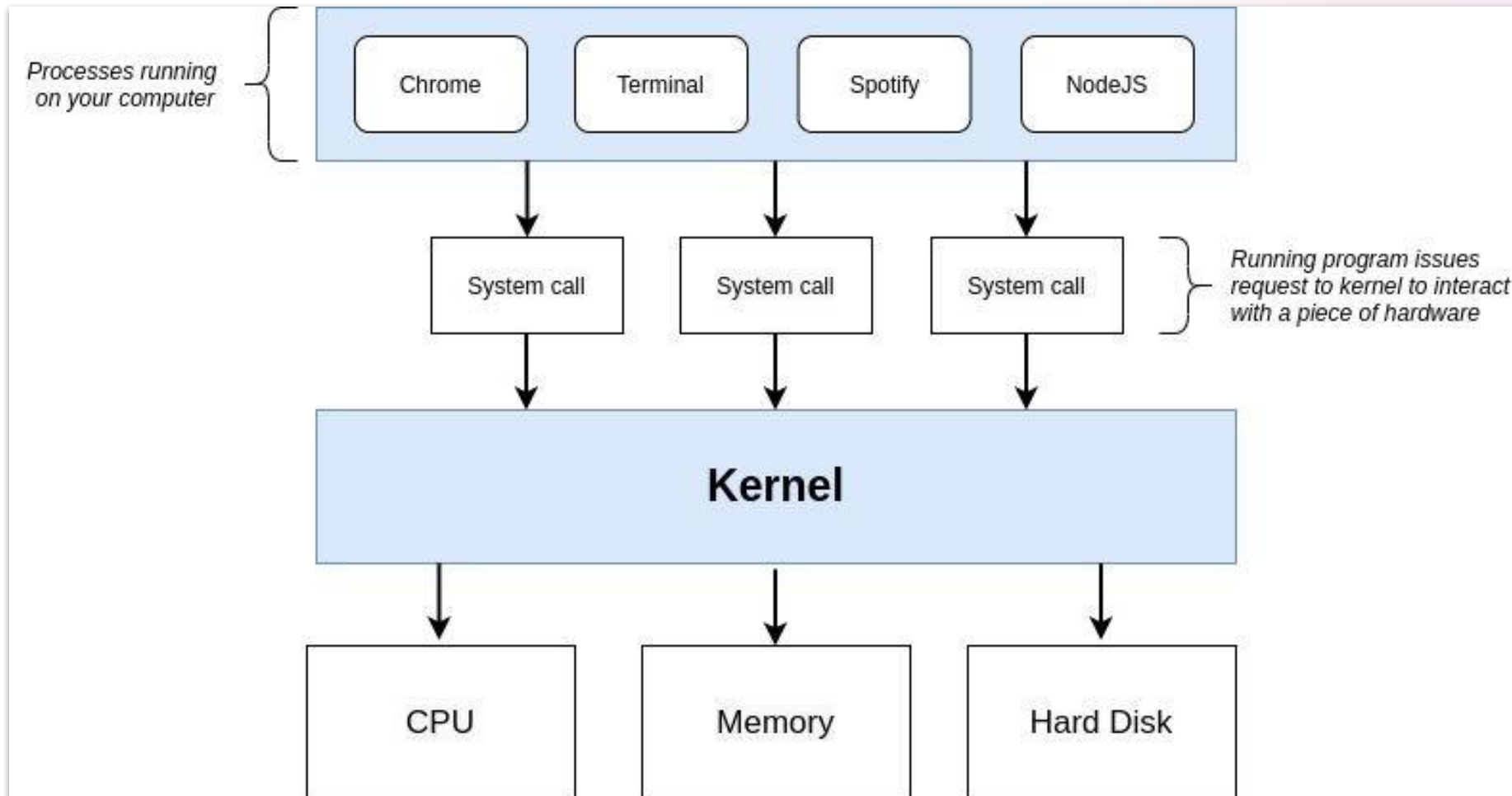
Install using the repository

- Set up the repository
 - `sudo apt-get update`
 - `sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release`
- Add Docker's official GPG key
 - `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg`
- Use the following command to set up the **stable** repository
 - `echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \ https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee \ /etc/apt/sources.list.d/docker.list > /dev/null`
- Install Docker engine
 - `sudo apt-get update`
 - `sudo apt-get install docker-ce docker-ce-cli containerd.io`
- Verify that Docker Engine is installed correctly by running the **hello-world** image.
 - `sudo docker run hello-world`

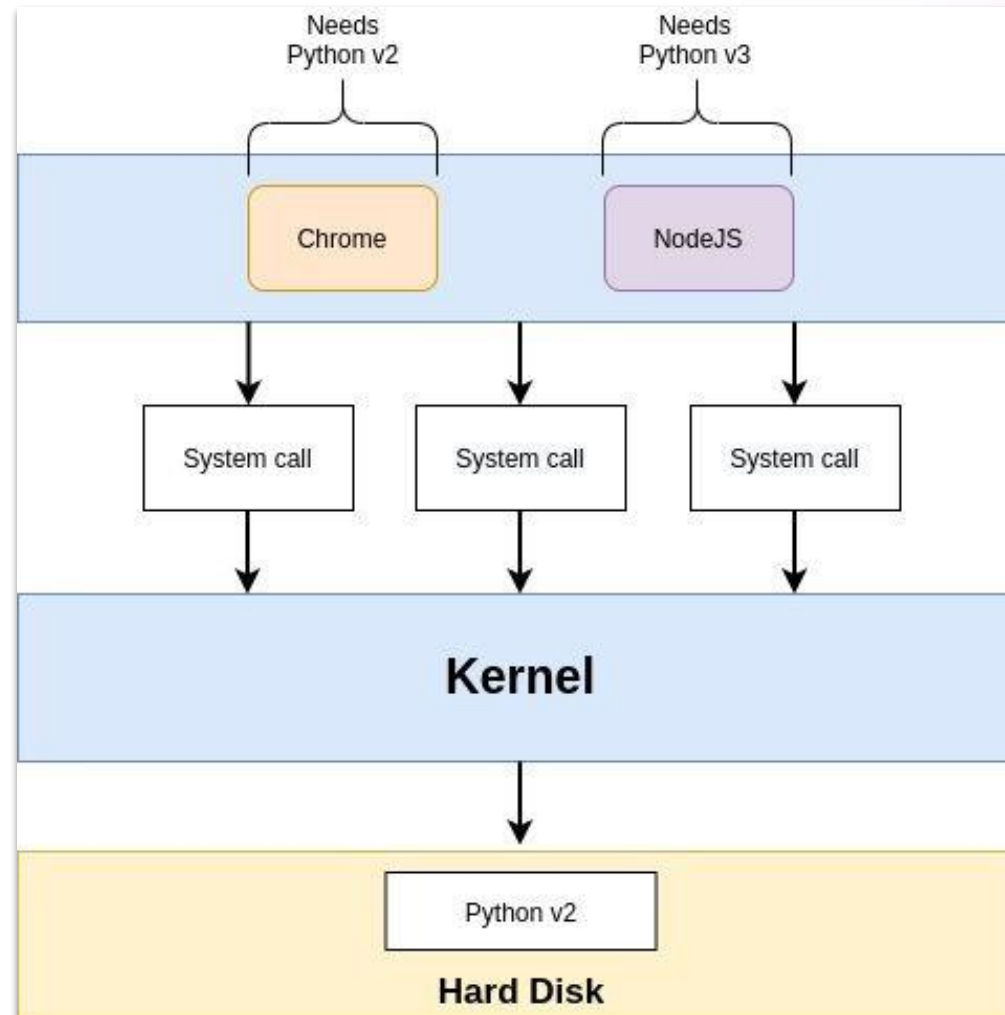
Process of *docker run hello-world* command



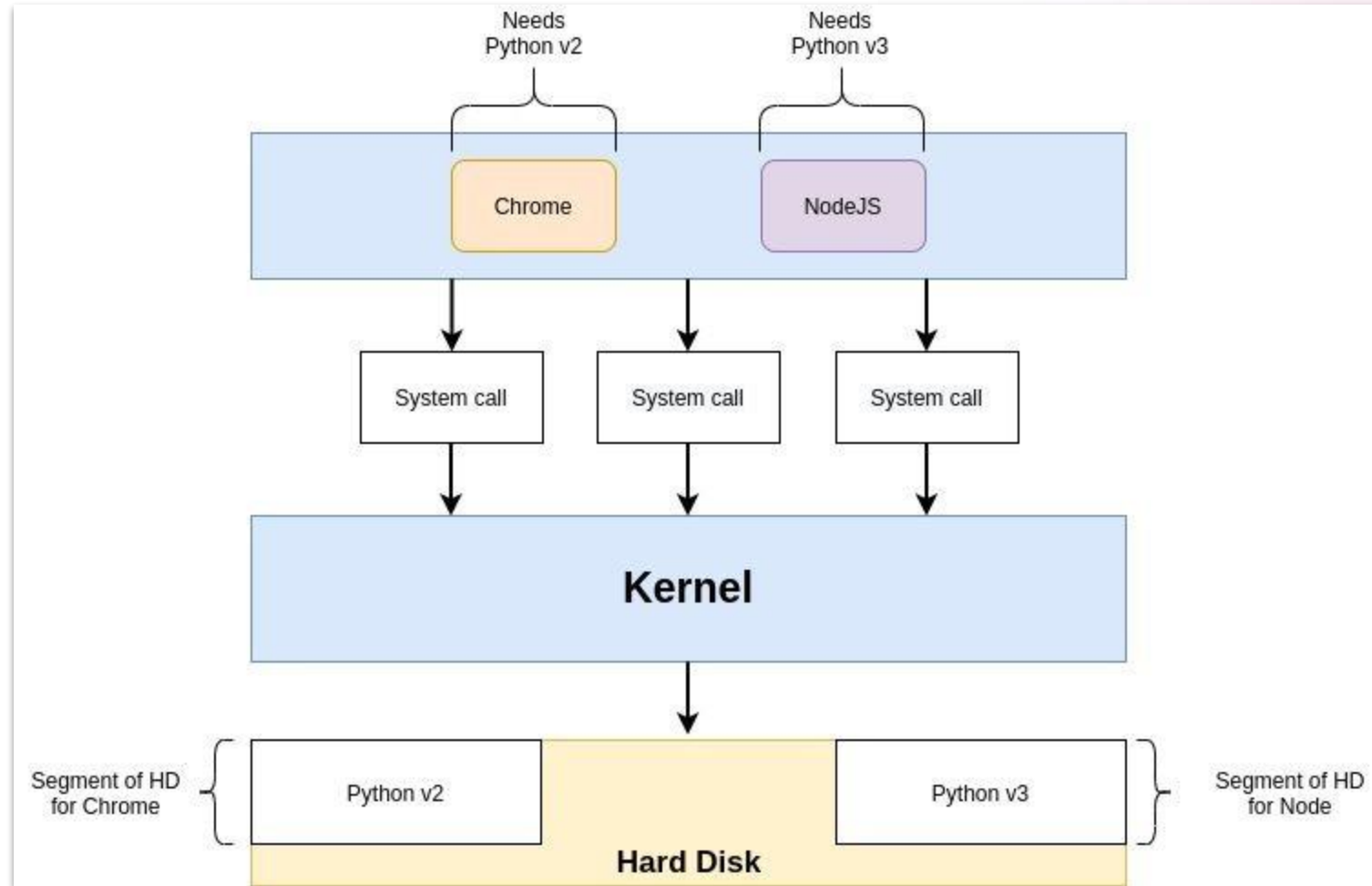
Quick overview of processes OS(Linux)



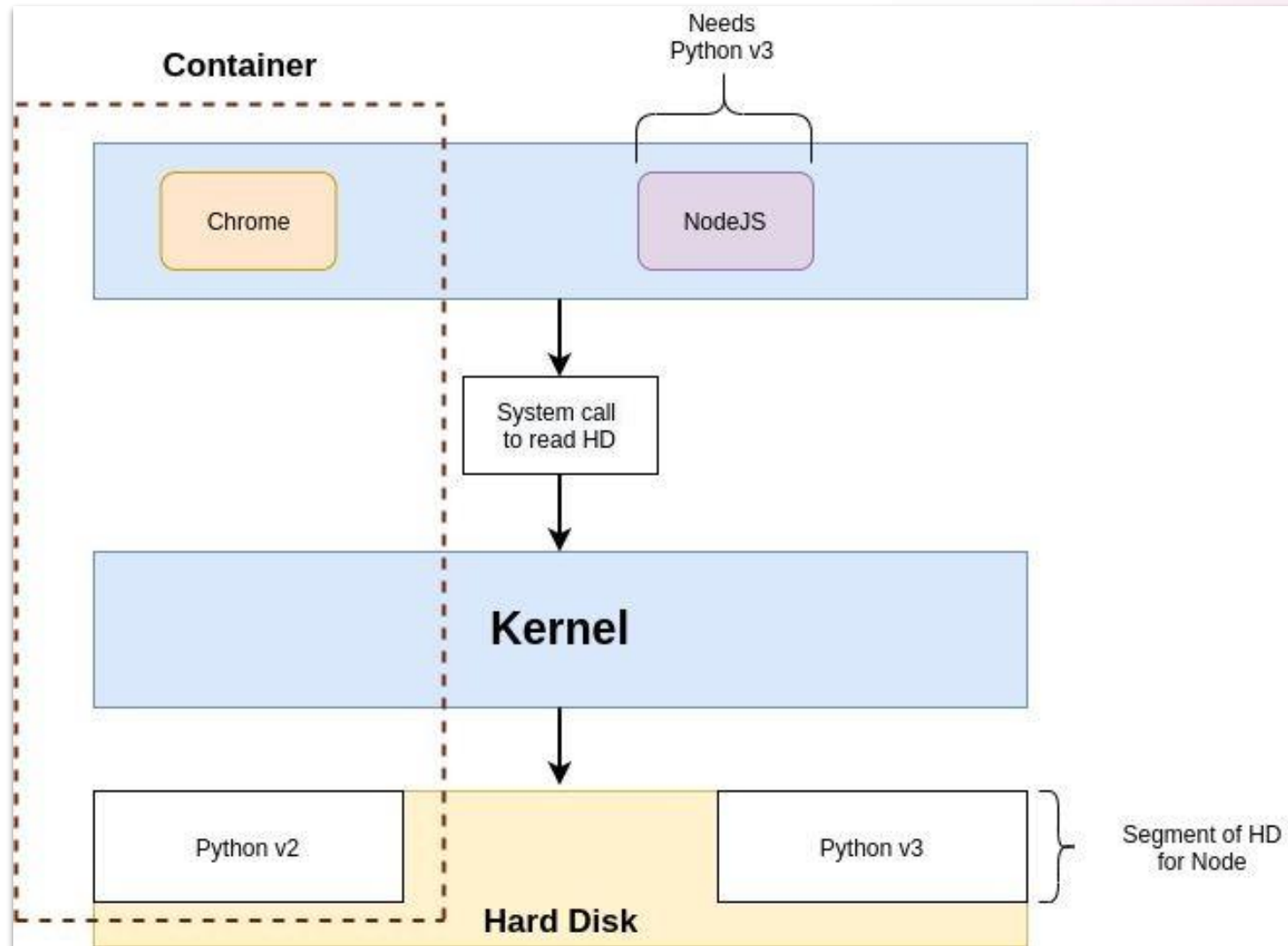
Quick overview of processes OS(Linux)



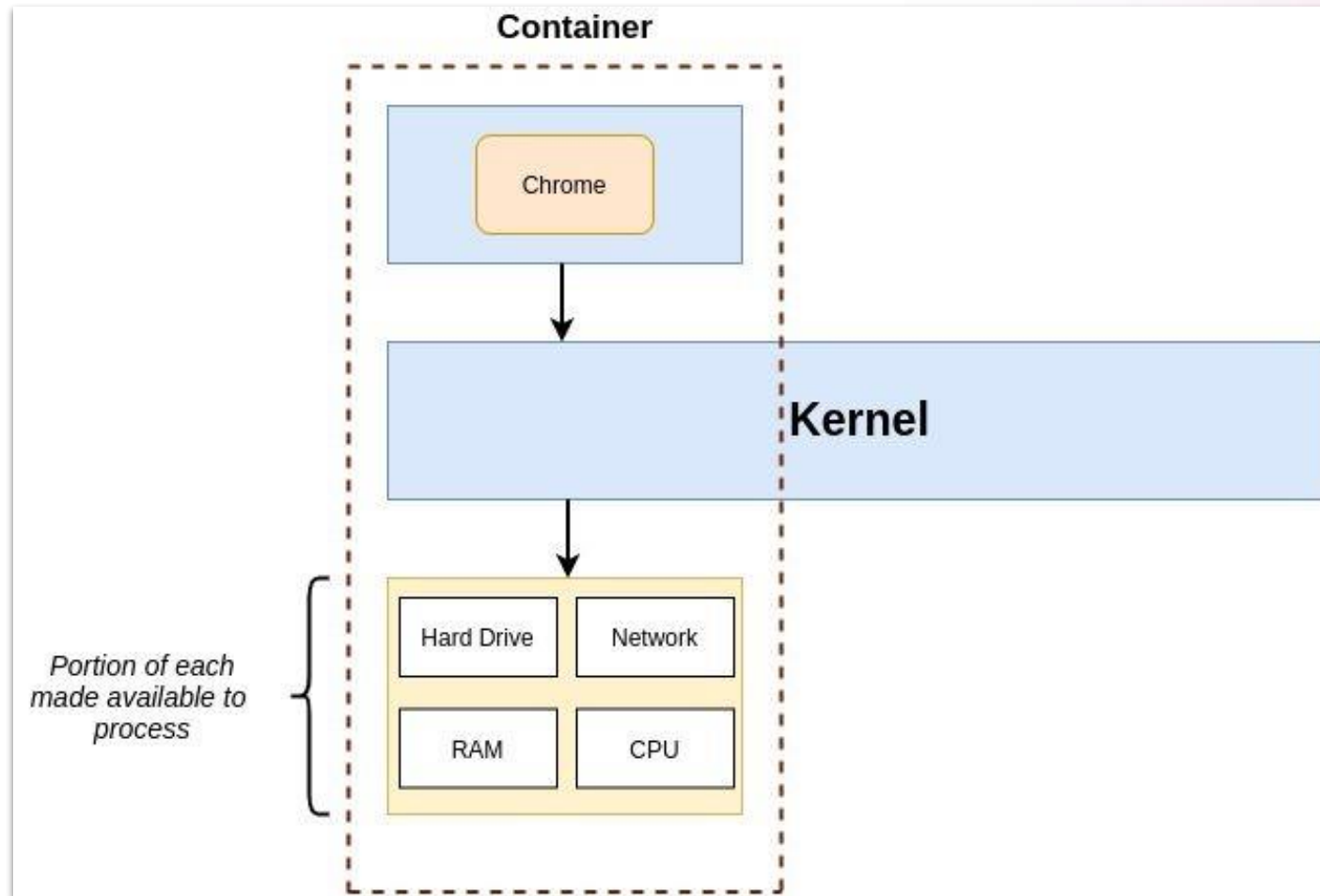
Quick overview of processes OS(Linux)



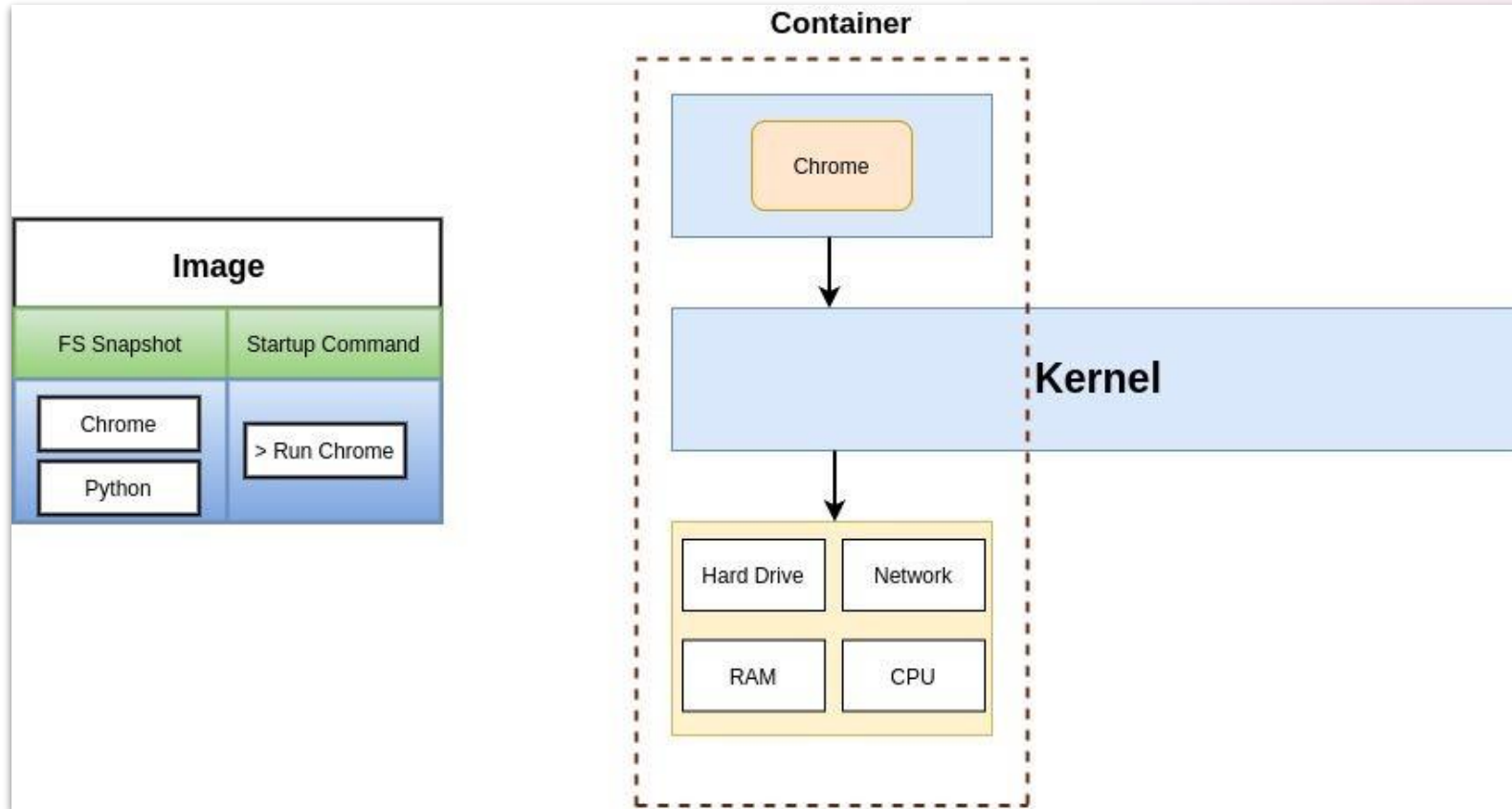
What is Container ?



What is Container ?

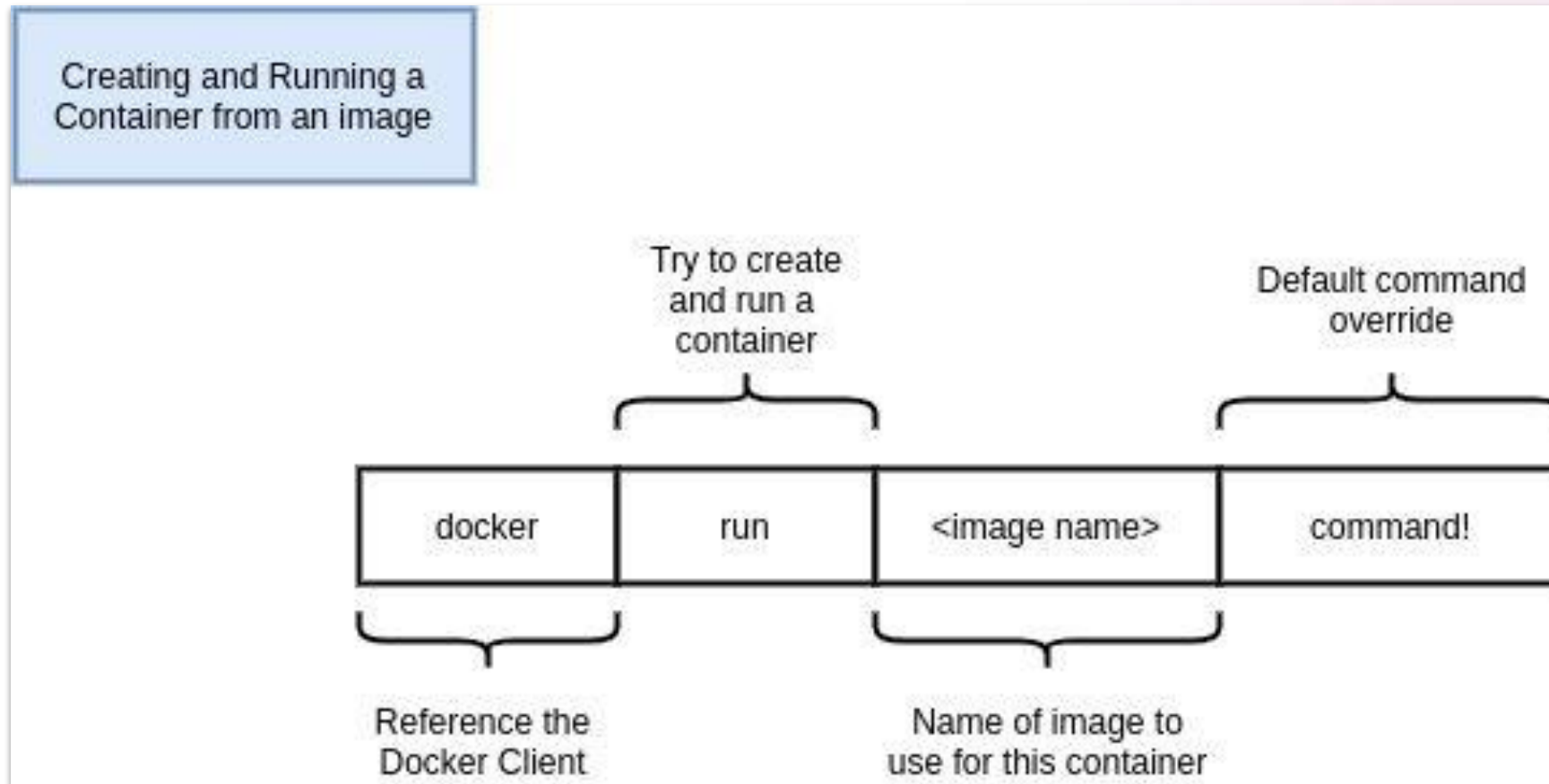


How image create Container

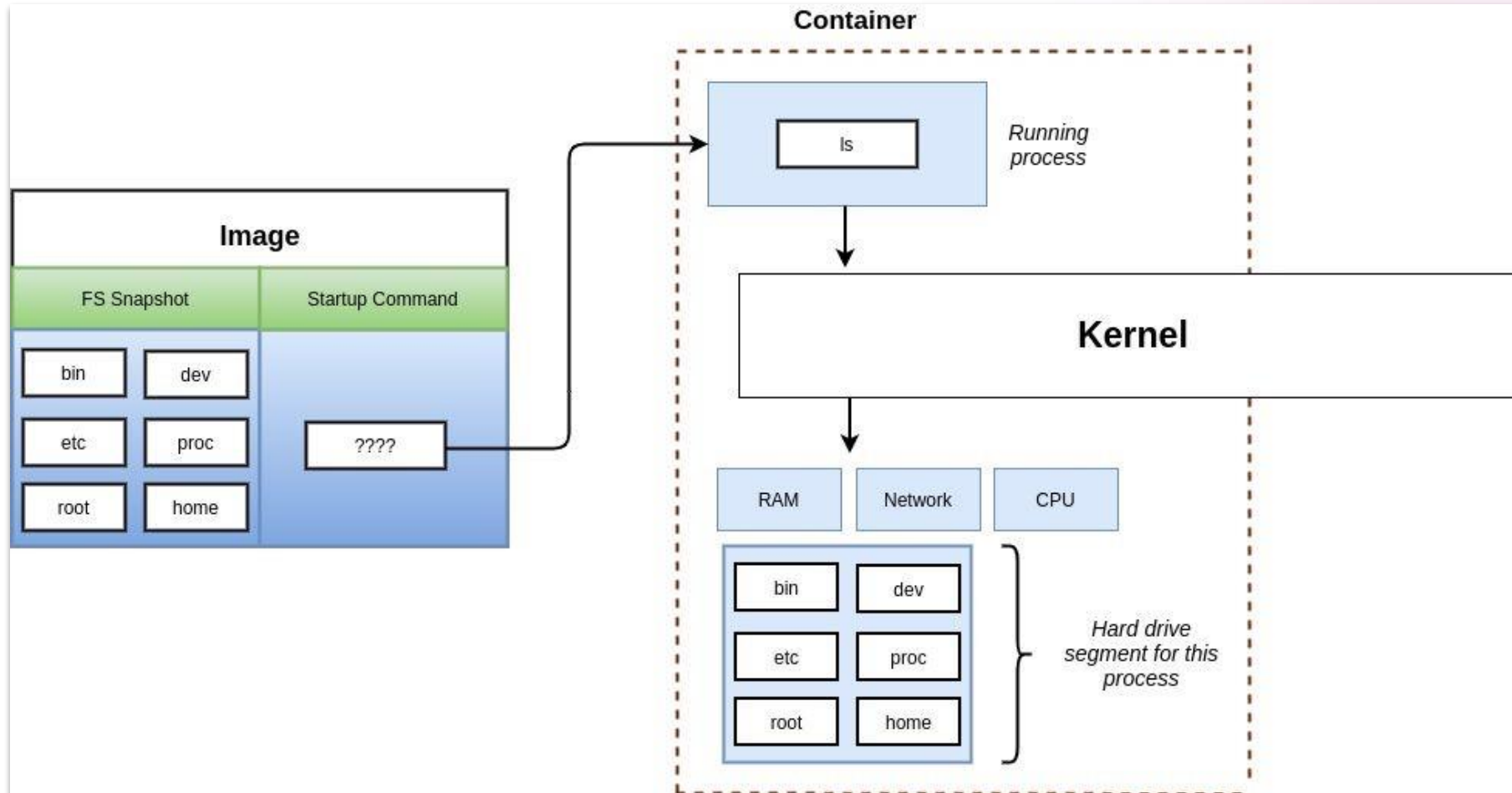


Manipulating Containers with the Docker Client

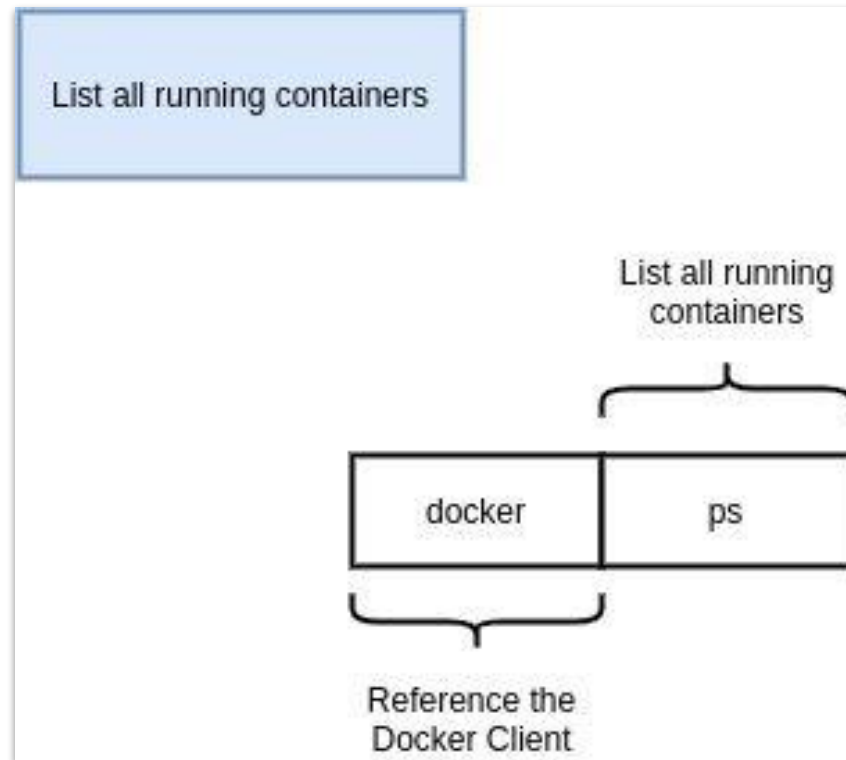
Overriding Default Commands



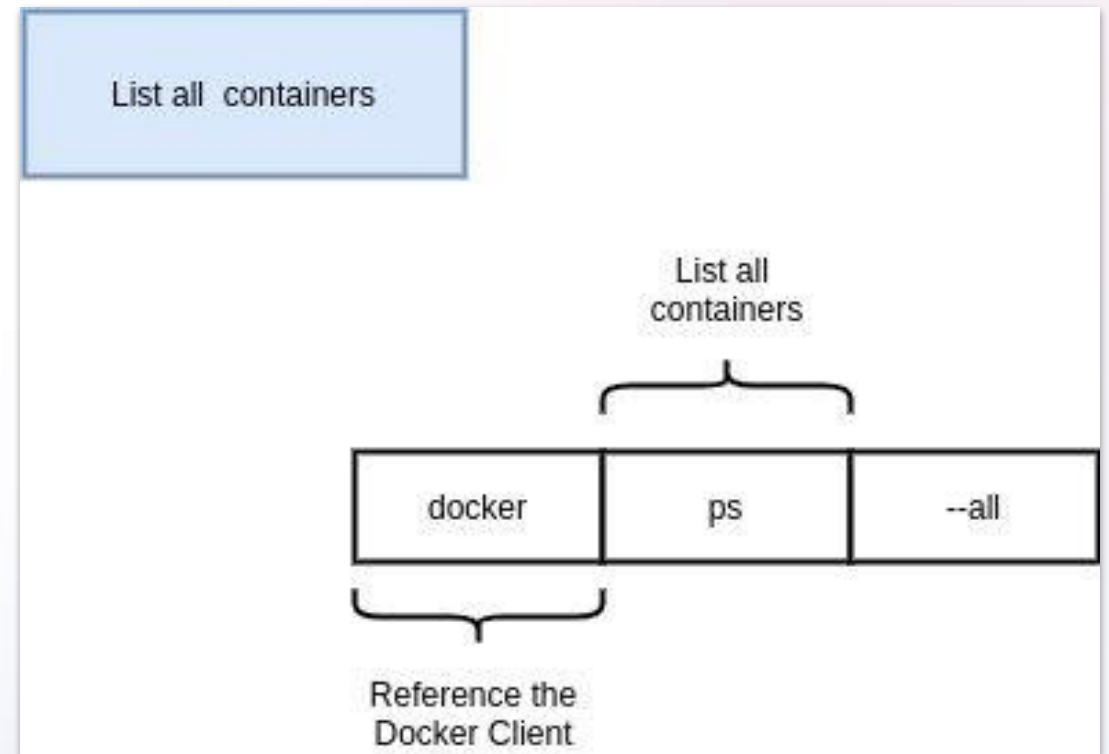
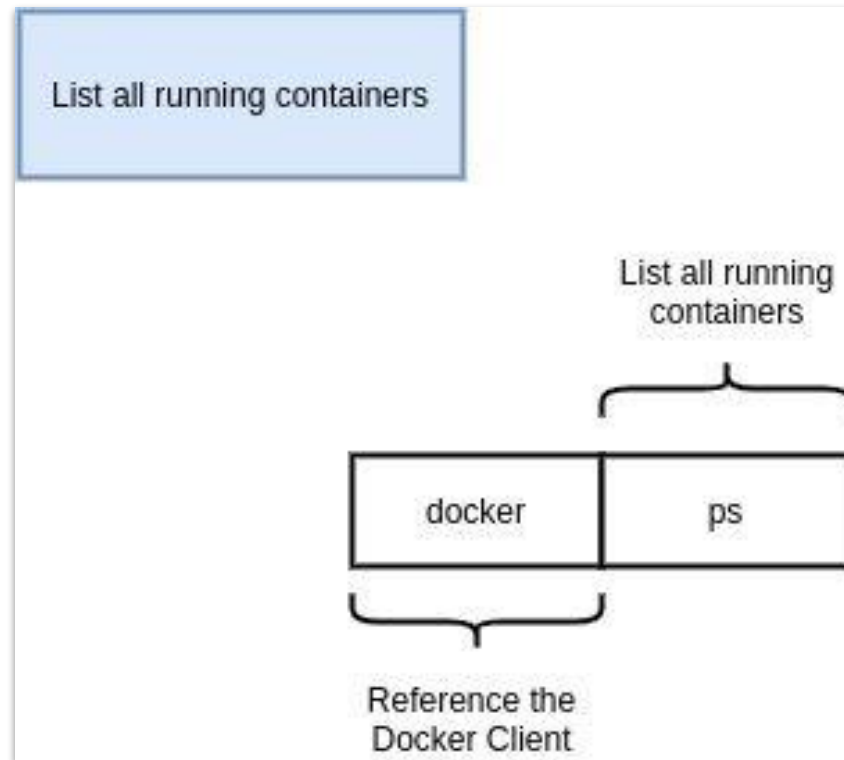
Overriding Default Commands



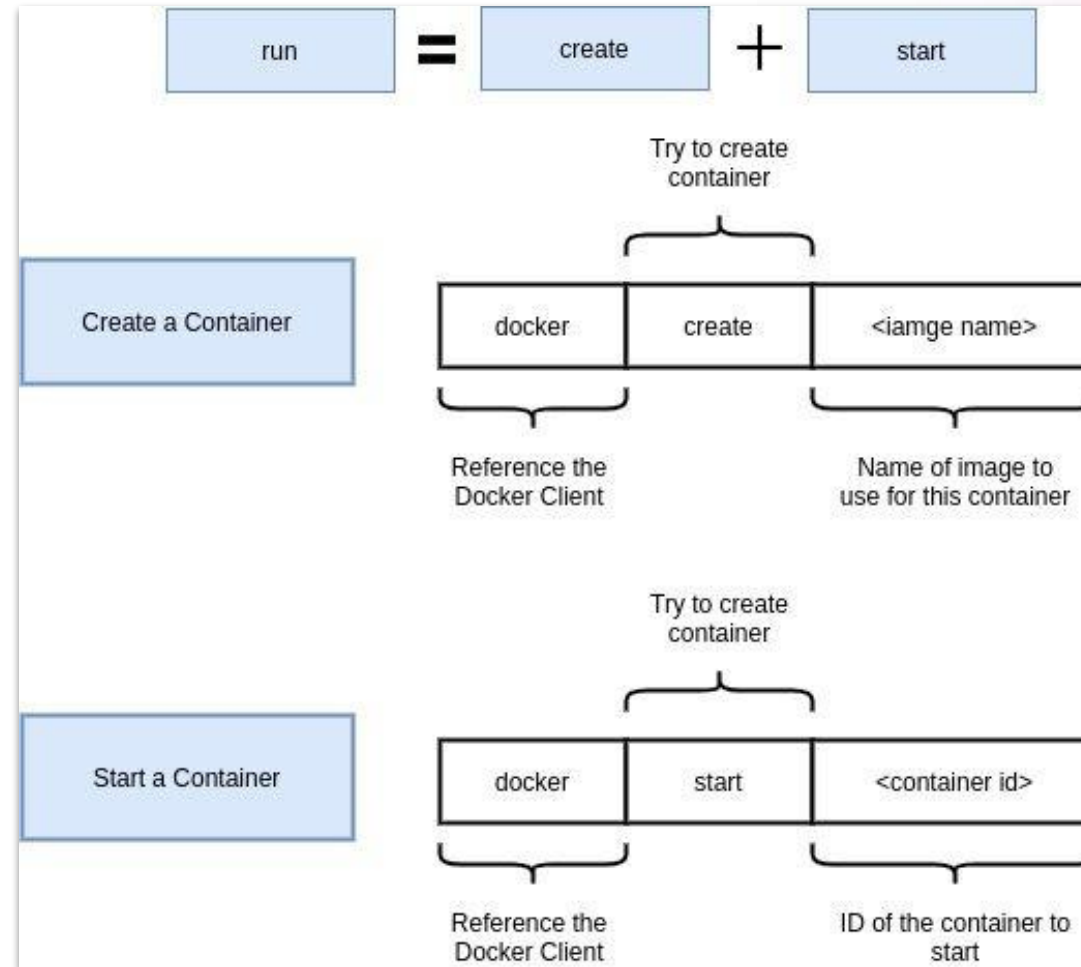
Overriding Default Commands



Overriding Default Commands



Overriding Default Commands



Overriding Default Commands

Removing stopped Containers

```
$ docker system prune
```

Removing one Container

```
$ docker rm <container id>/<container name>
```

Removing running Container

```
$ docker rm -f <container id>/<container name>
```

Retrieving log Outputs

```
$ docker logs <container id>/<container name>
```

Stopping Containers

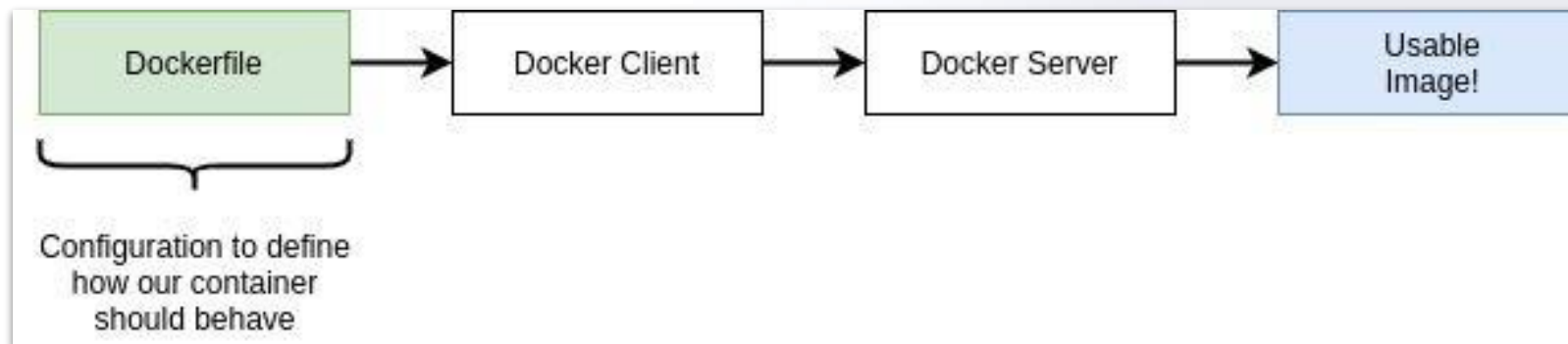
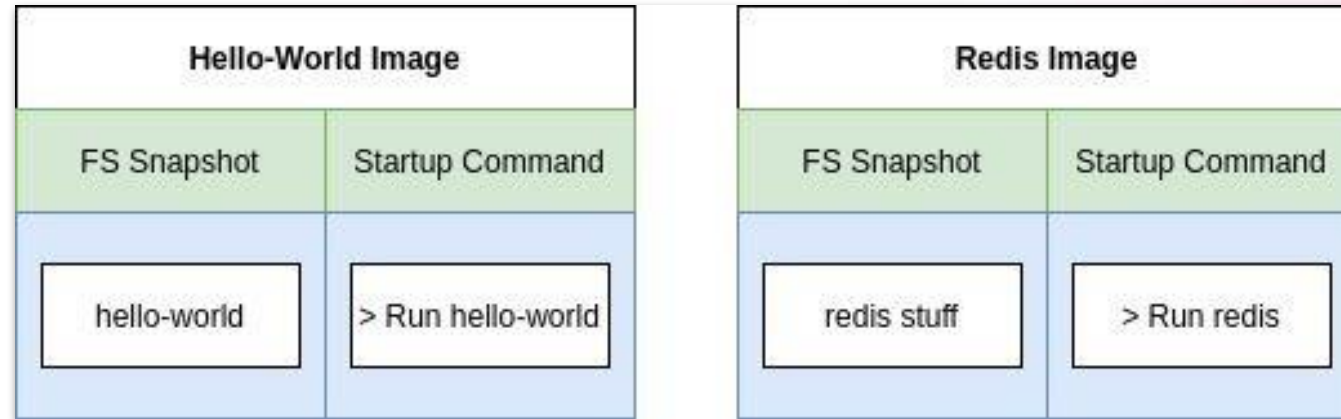
```
$ docker stop <container id>/<container name>
```

Killing Containers

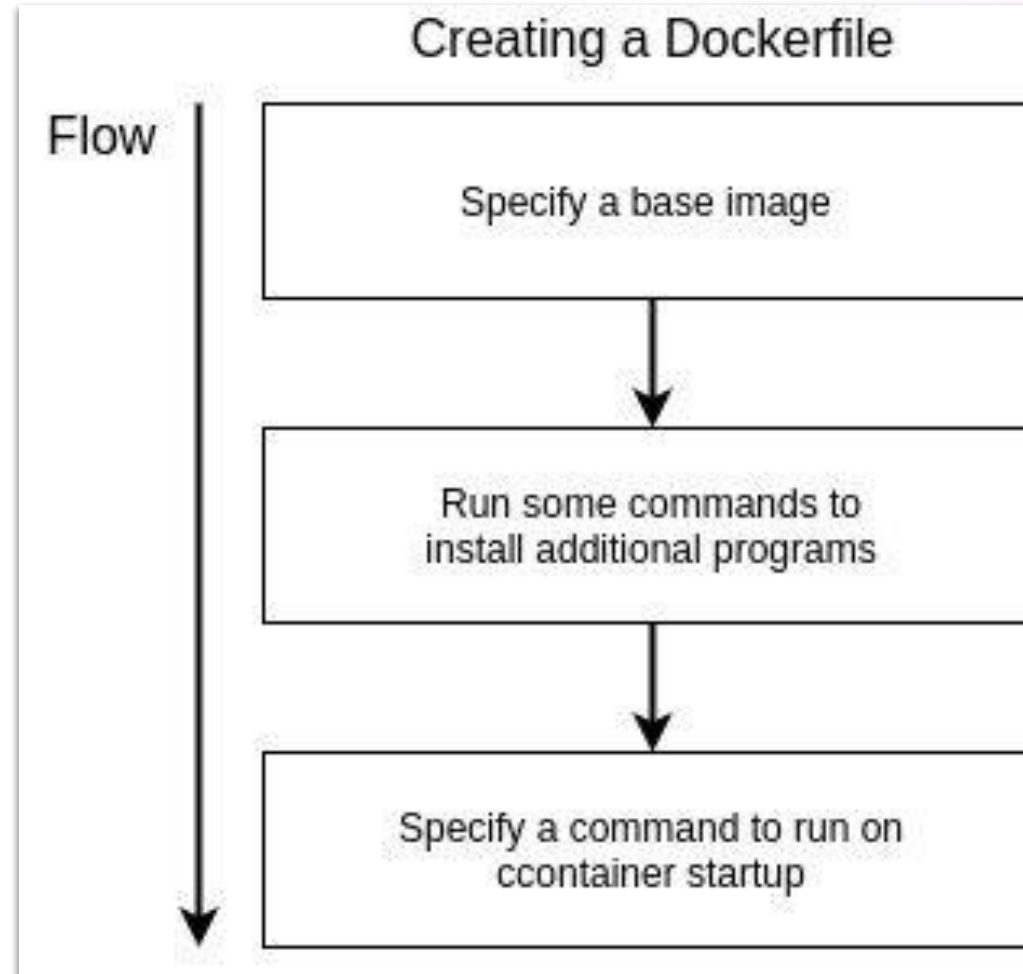
```
$ docker kill <container id>/<container name>
```

Build Custom Images

Flow of creating Custom Image



Flow of creating a DockerFile



Building a DockerFile

Create an image that runs redis-server

Goal

Use an existing docker image as a base

Download and install a dependency

Tell the image what to do when it start as a container

DockerFile Teardown

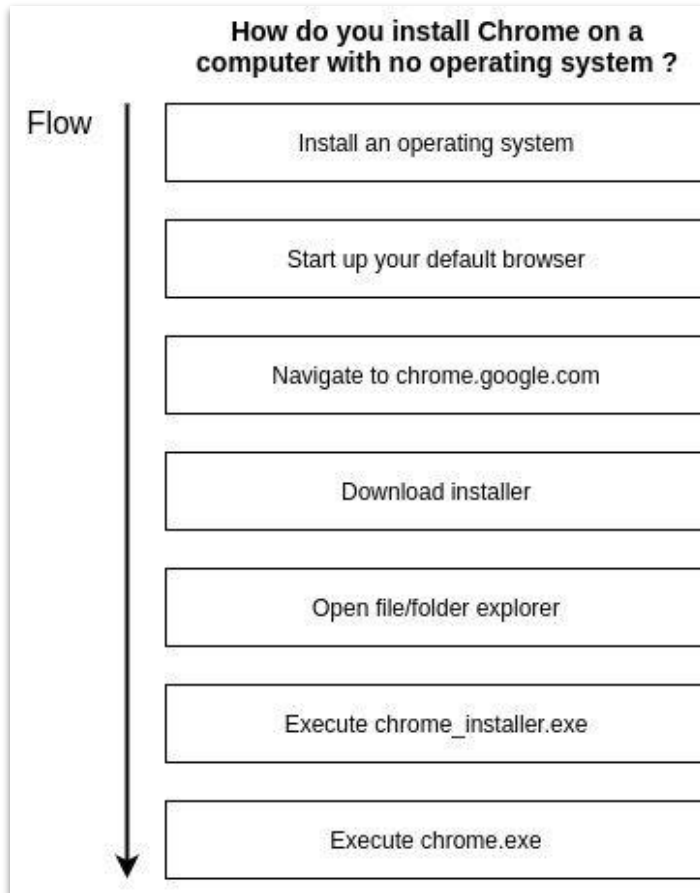
Instruction telling Docker Server what to do	Argument to the instruction
FROM	alpine
RUN	apk add --update
CMD	["redis-server"]

What's Base image

Imagine we have task to install Chrome with no OS in the computer

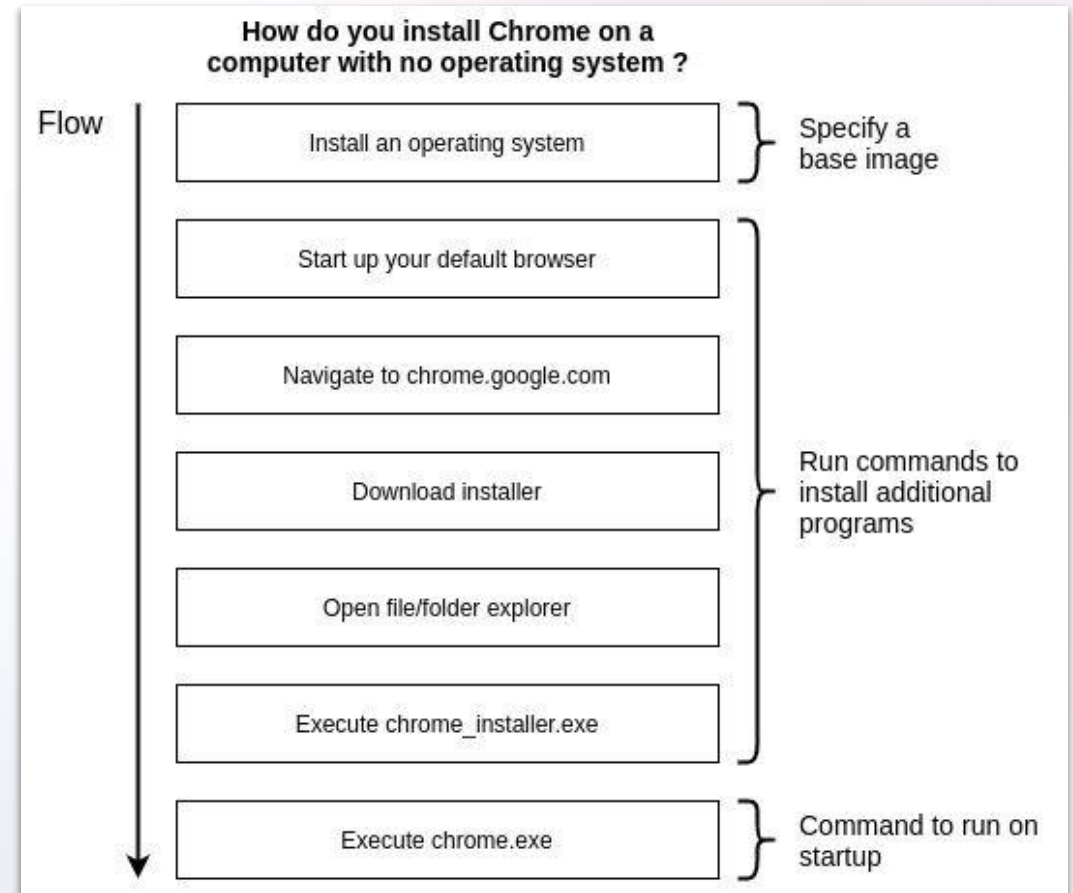
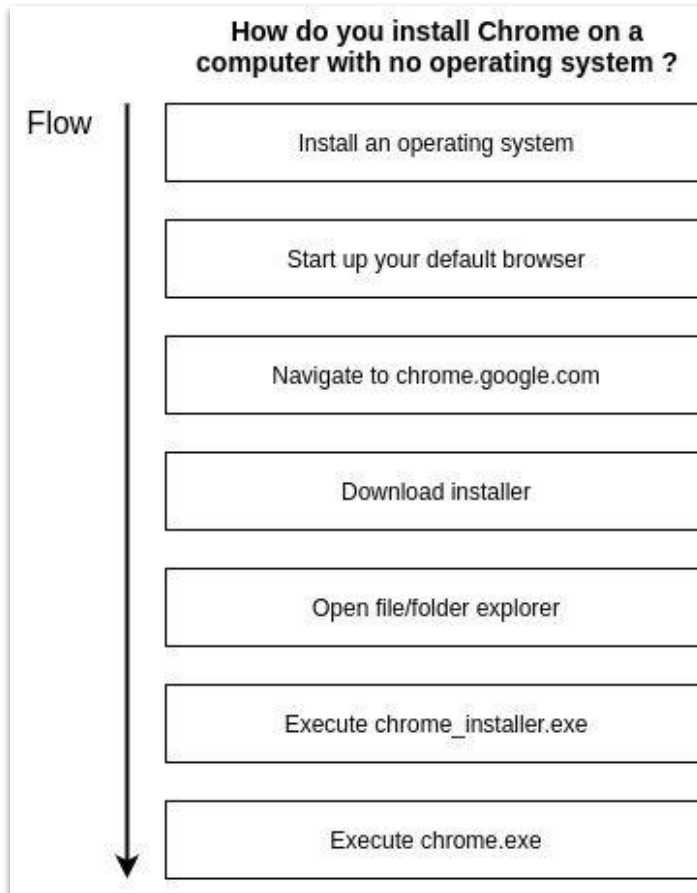
What's Base image

Imagine we have task to install Chrome with no OS in the computer

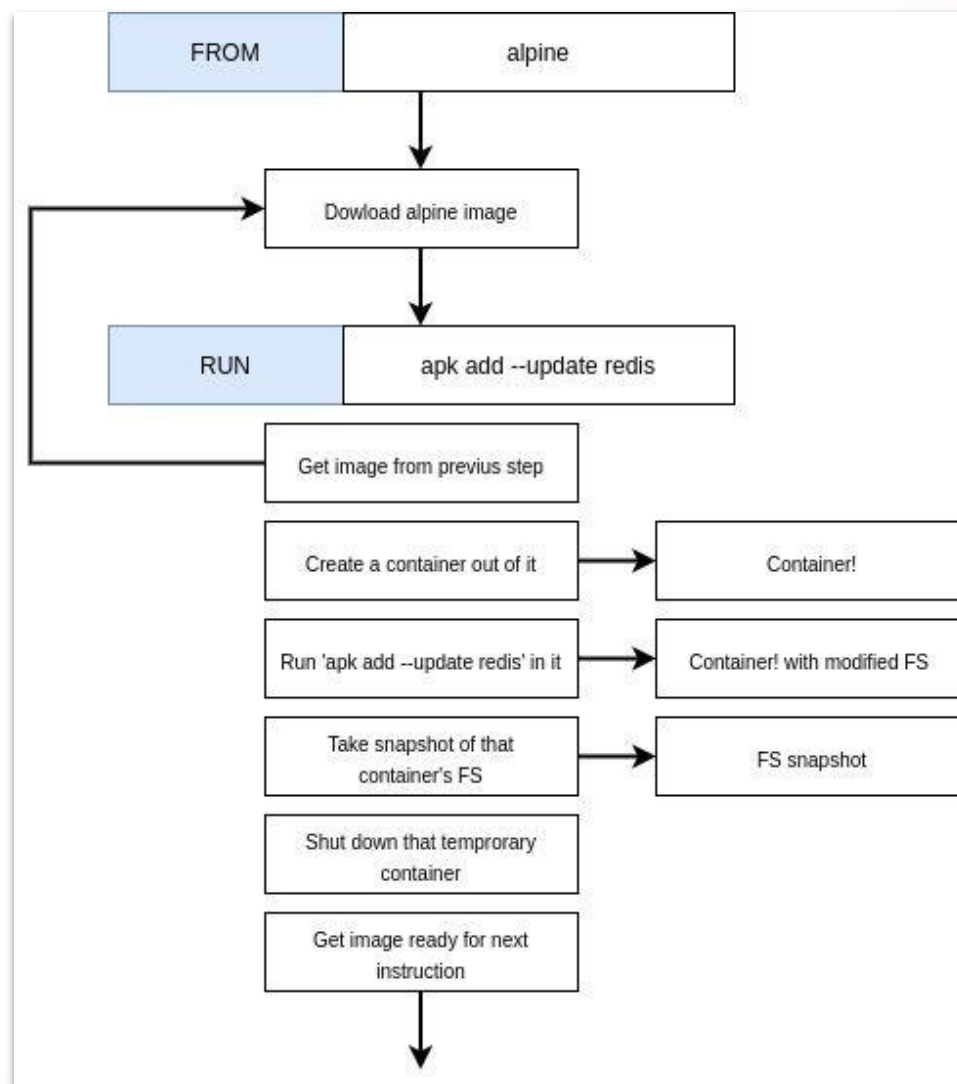


What's Base image

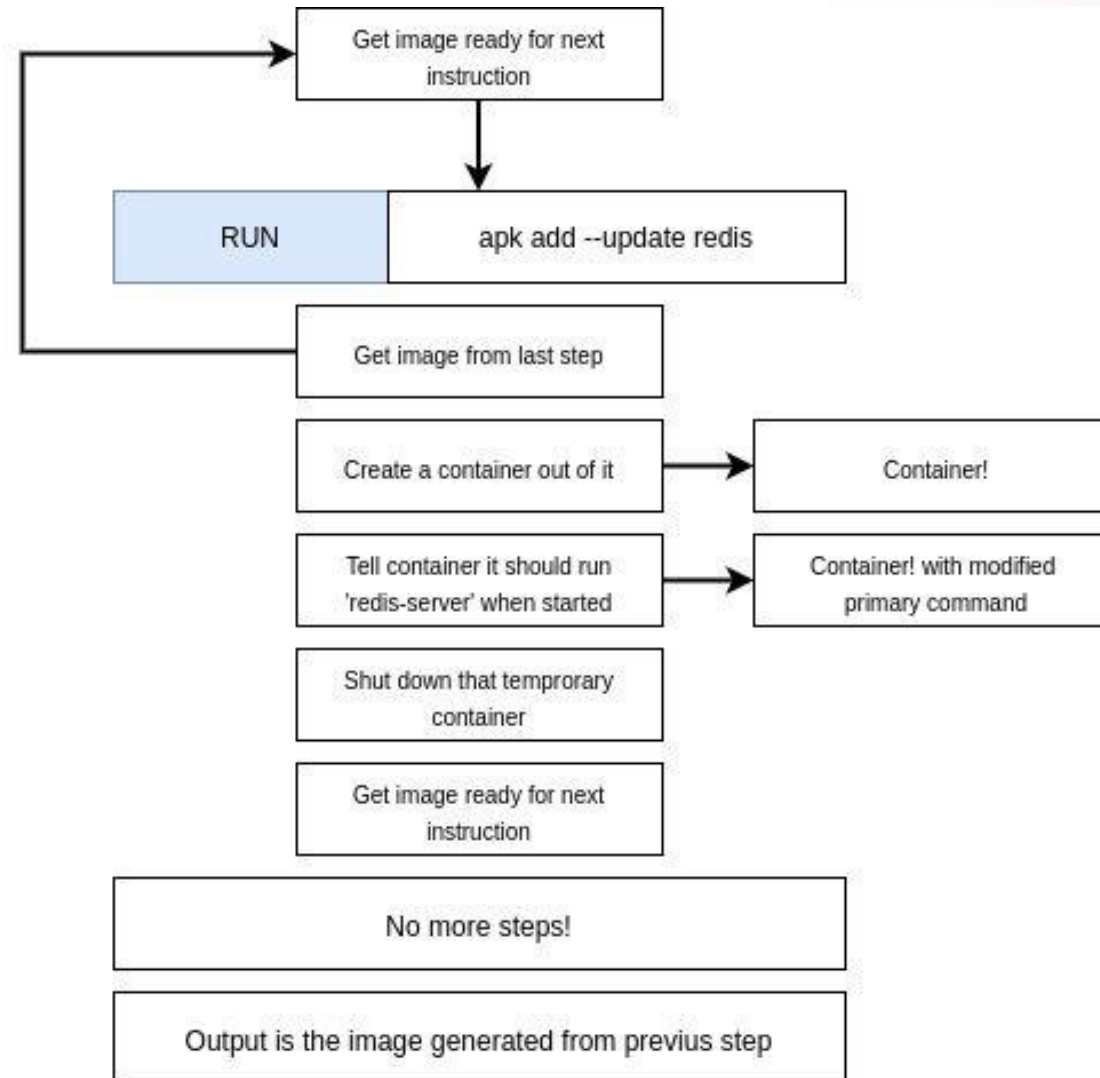
Imagine we have task to install Chrome with no OS in the computer



A brief Recap



A brief Recap



Rebuilds with cache

Example one

FROM	alpine
RUN	apk add --update redis
RUN	apk add --update curl
CMD	["redis-server"]

Example two

FROM	alpine
RUN	apk add --update curl
RUN	apk add --update redis
CMD	["redis-server"]

Building a DockerFile

Practical Work

Approximate Duration: 15 minutes

Objectives:

Create custom image from Dockerfile

Thank you for your attention !

Q&A

