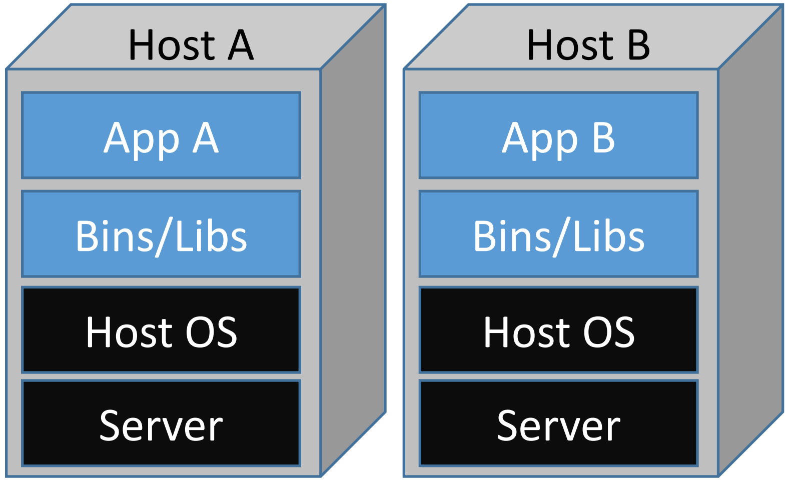
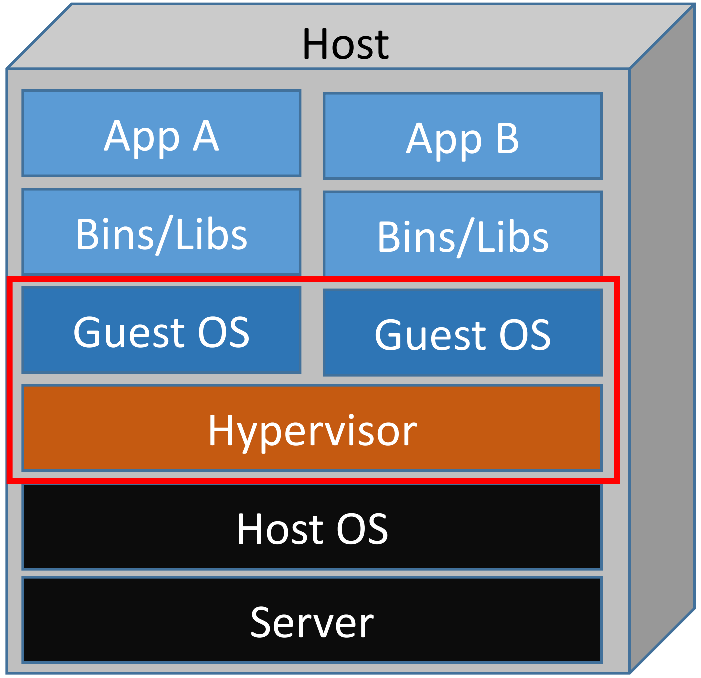
Linux Container (LXC)

- **Linux Container (LXC)** is an operating-system-level virtualization method
- Multiple isolated Linux systems run in a control host using single Linux kernel
- Each Container has its "own" isolated runtime environment:
  - file system
  - network
  - process tree
- It is faster and more portable than using *Virtual Machines*

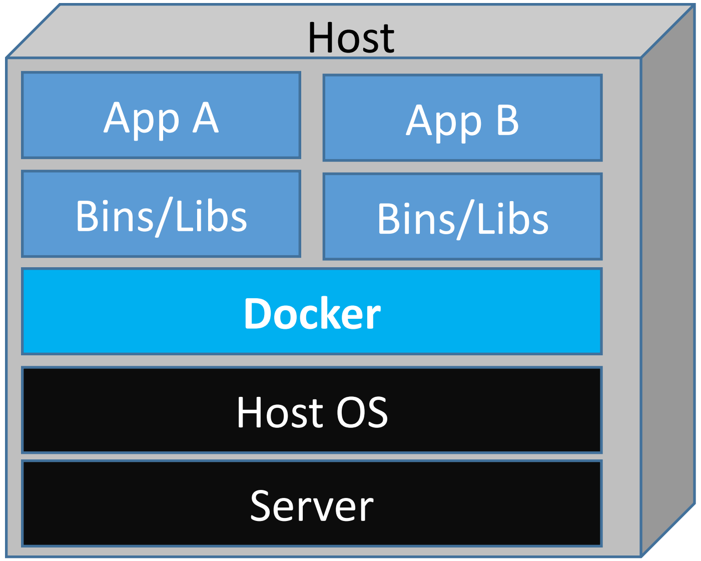
Multiple Hosts



Virtual Machines

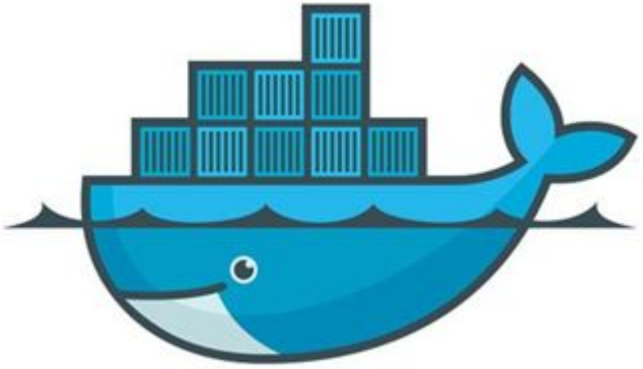


Containers



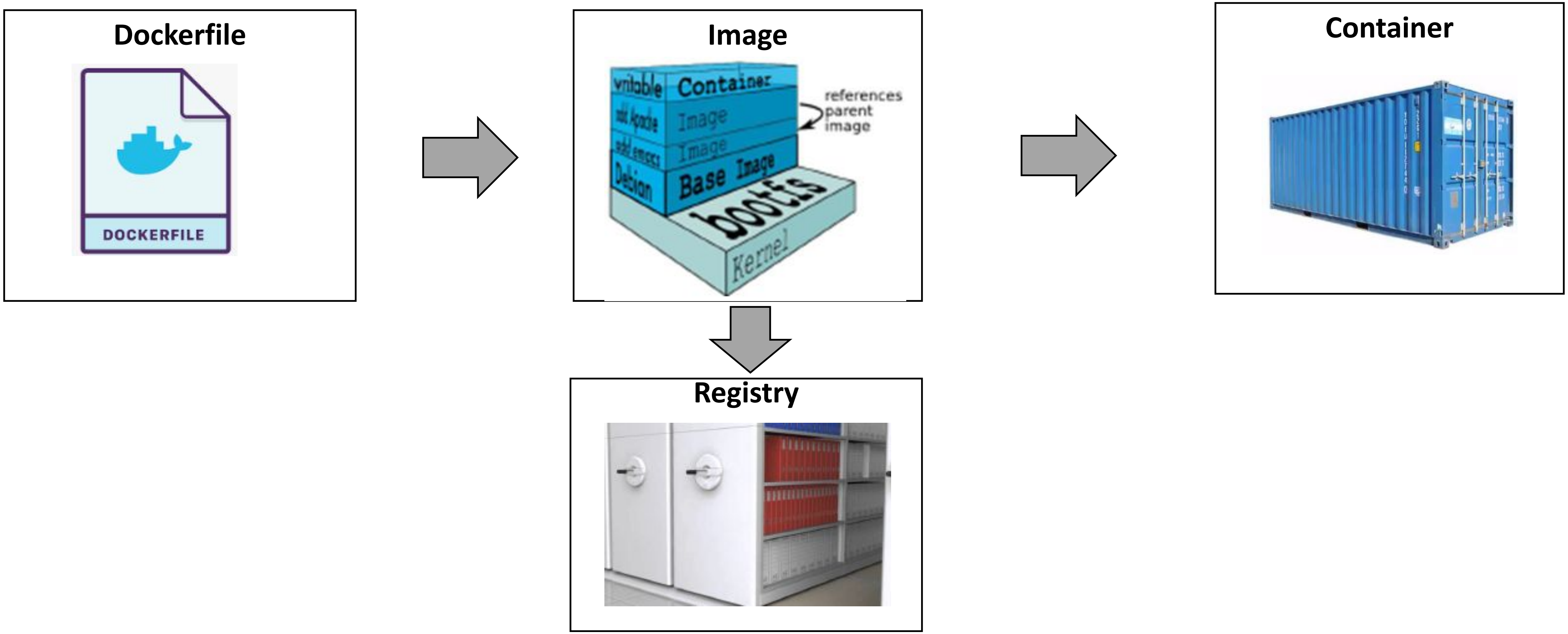
Docker

- The **Docker** is an actual implementation of Containers technology.
- It was created in 2013 by Docker (former dotCloud) company
- Supported platforms:
    - Linux, Mac, Windows-10, WinServer-2016 – Native Application
    - Windows < 10 – Docker Toolbox



Docker Architecture

- **Docker Engine** is daemon service installed and running on OS
- **Docker Images** are multi-layer templates used for creation of Containers
- **Dockerfile** configuration file with textual instructions how to build the Docker Image
- **Docker Containers** are running instances of Docker Image – Linux Application running in Docker environment
- **Docker Registries** – the storages of Docker Images. Most known is **Docker Hub**



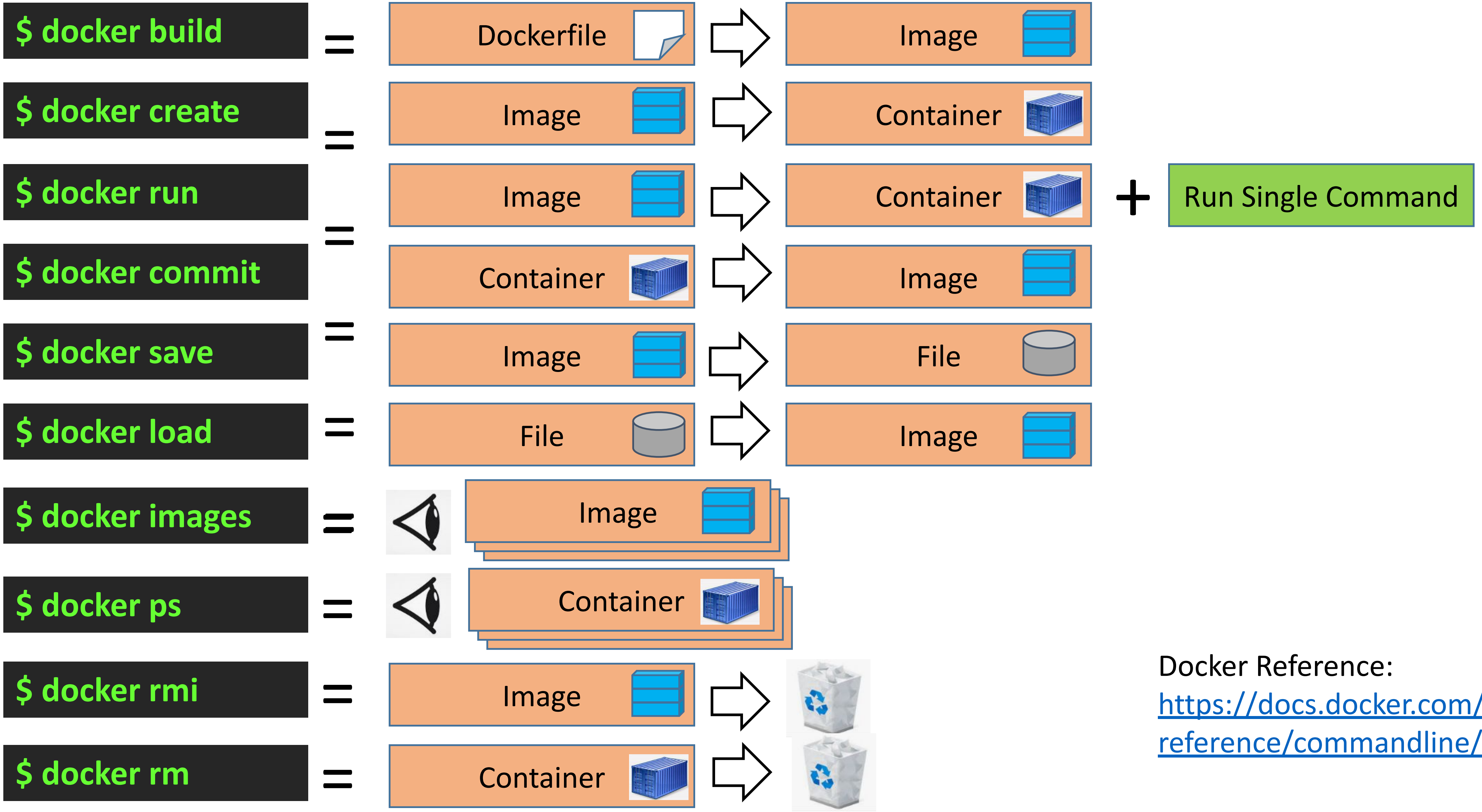
Dockerfile

Image

Container

Registry

Docker Basic Commands



$ docker build = Dockerfile → Image

$ docker create = Image → Container

$ docker run = Image → Container + Run Single Command

$ docker commit = Container → Image

$ docker save = Image → File

$ docker load = File → Image

$ docker images = Image

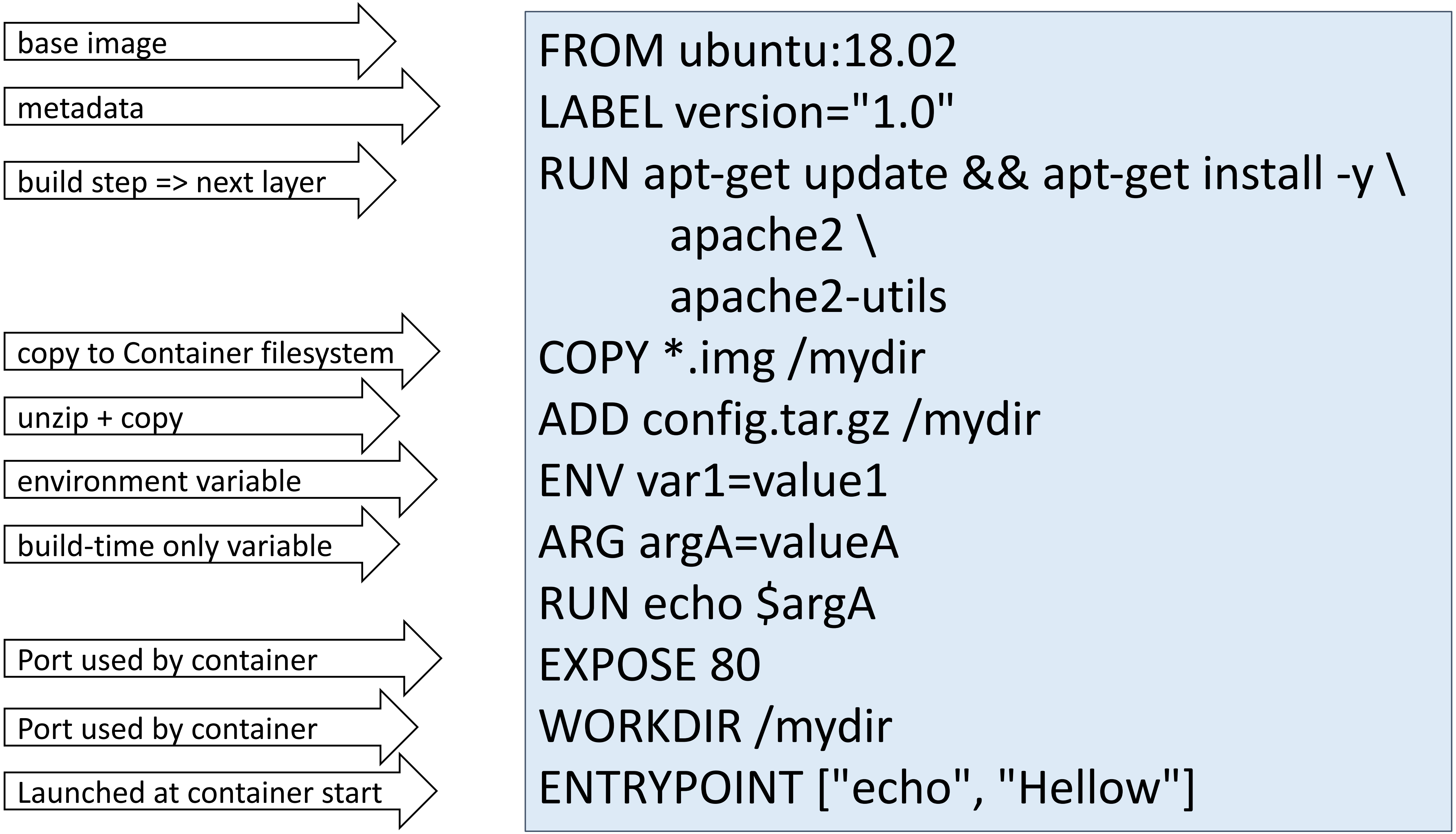
$ docker ps = Container

$ docker rmi = Image →

$ docker rm = Container →

Docker Reference:
https://docs.docker.com/engine/reference/commandline/cli/

Dockerfile Syntax

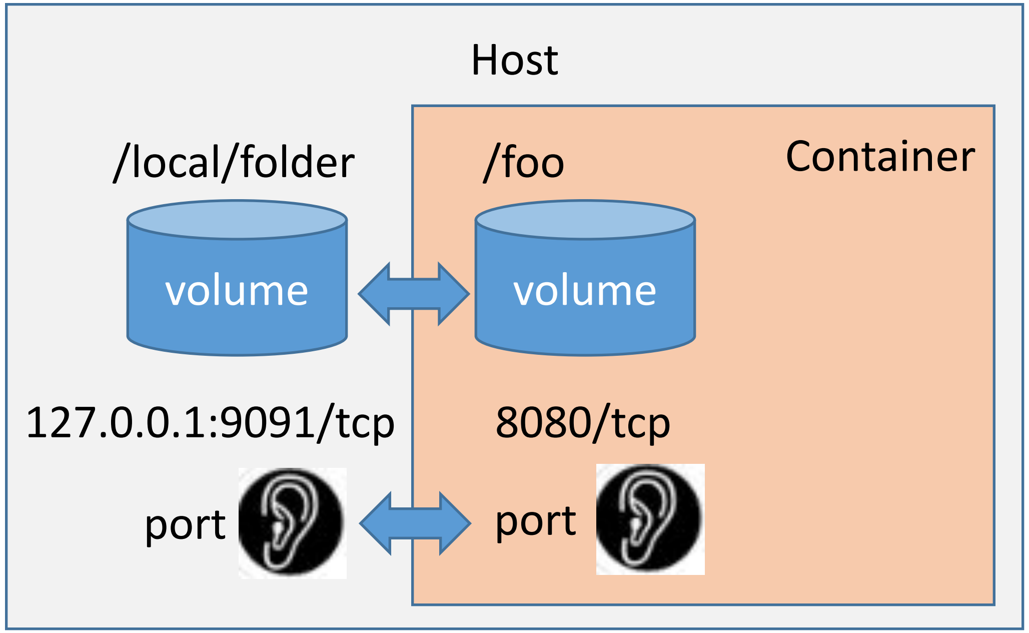


- base image
- metadata
- build step => next layer
- copy to Container filesystem
- unzip + copy
- environment variable
- build-time only variable
- Port used by container
- Port used by container
- Launched at container start

```
FROM ubuntu:18.02
LABEL version="1.0"
RUN apt-get update && apt-get install -y \
          apache2 \
          apache2-utils
COPY *.img /mydir
ADD config.tar.gz /mydir
ENV var1=value1
ARG argA=valueA
RUN echo $argA
EXPOSE 80
WORKDIR /mydir
ENTRYPOINT ["echo", "Hellow"]
```

Docker Run: Volume and Port Mappings

```
docker run -v /local/folder:/foo \          # volume mapping
      -w /foo  \                           # working dir
      -p 127.0.0.1:9091:8080/tcp \         # port mapping
      -it \                         # interactive shell
            my_image \                 # image
      bash                          # run in container
```

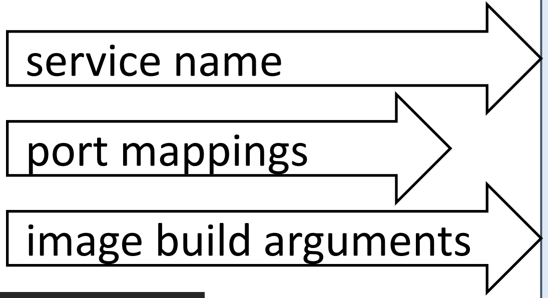


Tool docker-compose

The **Docker Compose** is the tool for defining and running multi-container applications with Docker

- Defines multi-container application in singe YML file
- Spin the application in single command

`



service name

port mappings

image build arguments

```
version: '2.1'
services:
 programmer-email-provider:
  ports:
  - "9393:9393"
  build:
   logs-bugs-email-provider

 assigner-email-provider:
  ports:
  - "9292:9292"
  build:
   logs-bugs-assigner-mail-provider

 email-notifier:
  build:
   logs-bugs-email-notifier
```

```
$ docker-compose build    # build images
$ docker-compose up -d    # start (detached)
$ docker-compose logs     # show logs
$ docker-compose ps # show containers
$ docker-compose down     # stop and remove
$ docker-compose scale    # set number of containers
```

Documentation:
https://docs.docker.com/compose/

See Also

Play with Docker – interactive playground https://labs.play-with-docker.com/

Portainer.io – Docker-based container manager with web interface: https://www.portainer.io/

Docker Swarm mode – managing a cluster of Docker Engines: https://docs.docker.com/engine/swarm/