Docker Deep Dive

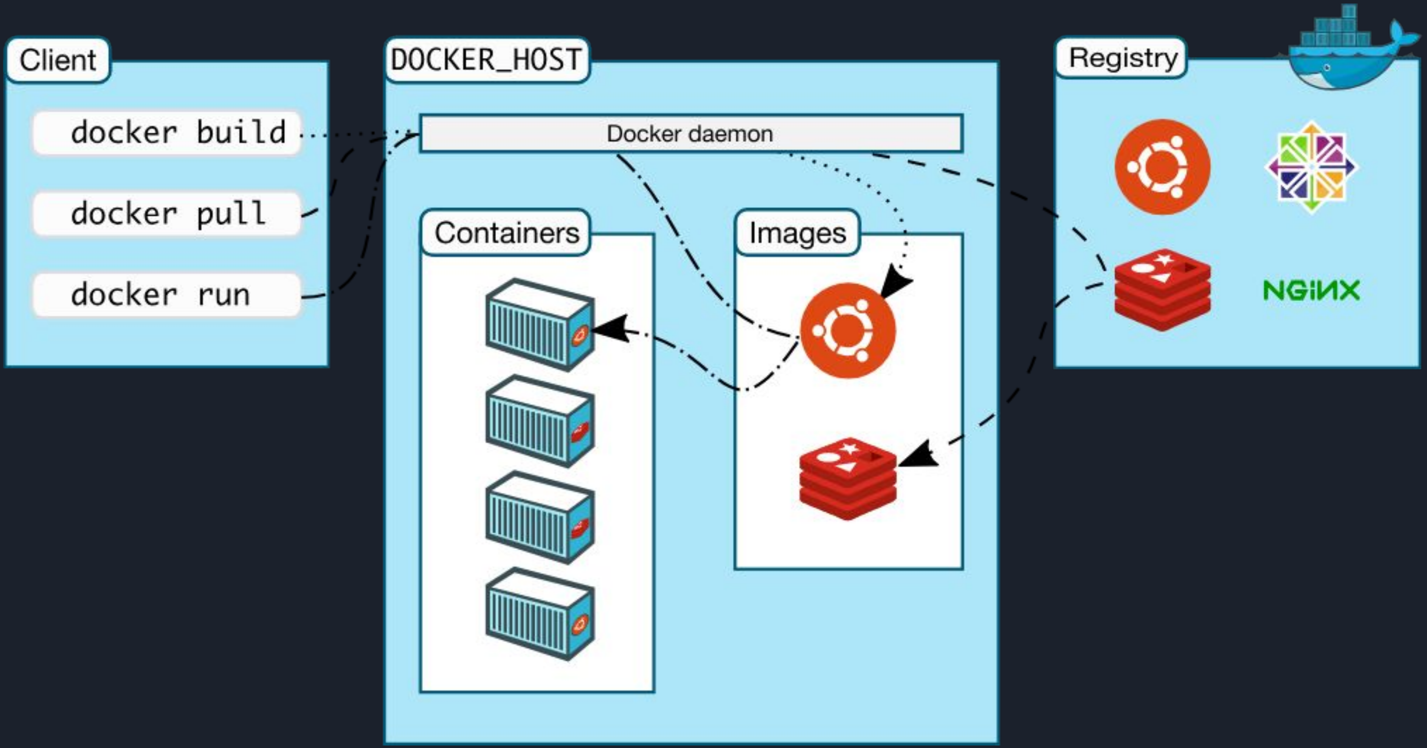
Docker WorkShop

By DTherHtun

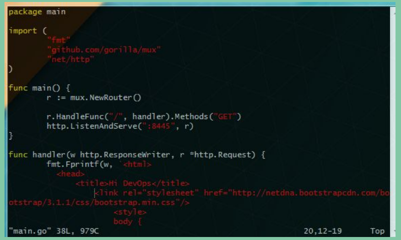
Agenda

- Docker Architecture
- Docker Installing
- Docker Image layers
- A little play with docker engine and containers
- Docker Build
- Docker Registry
- Docker Networking
- Docker Troubleshooting

Docker Architecture

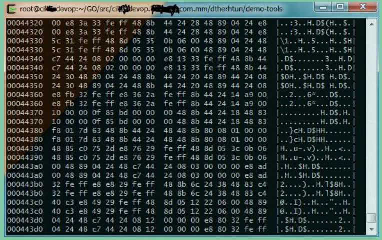


Program



```
package main

import (
        "fmt"
        "github.com/gorilla/mux"
        "net/http"
)

func main() {
        r := mux.NewRouter()

        r.HandleFunc("/", handler).Methods("GET")
        http.ListenAndServe(":8445", r)
}

func handler(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintf(w, `<html>
          <head>
              <title>Hi DevOps</title>
                  <link rel="stylesheet" href="http://netdna.bootstrapcdn.com/bo
otstrap/3.1.1/css/bootstrap.min.css"/>
                        <style>
                        body {
"main.go" 38L, 979C                                         20,12-19      Top
```

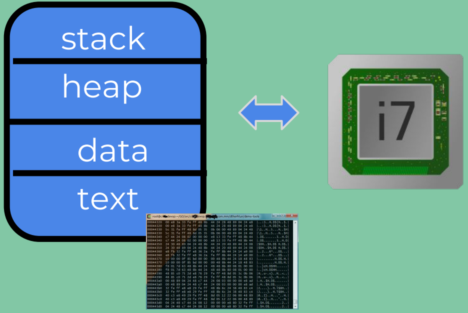
sourcecode



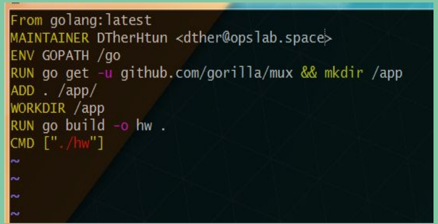
```
root@ci[redacted]devop:~/GO/src/ci[redacted]devop.[redacted]com.mm/dtherhtun/demo-tools
00044320  00 e8 3a 33 fe ff 48 8b  44 24 28 48 89 04 24 e8  |..:3..H.D$(H..$.|
00044320  00 e8 3a 33 fe ff 48 8b  44 24 28 48 89 04 24 e8  |..:3..H.D$(H..$.|
00044330  5c 31 fe ff 48 8d 05 35  0b 06 00 48 89 04 24 48  |\1..H..5...H..$H|
00044330  5c 31 fe ff 48 8d 05 35  0b 06 00 48 89 04 24 48  |\1..H..5...H..$H|
00044340  c7 44 24 08 02 00 00 00  e8 13 33 fe ff 48 8b 44  |.D$.......3..H.D|
00044340  c7 44 24 08 02 00 00 00  e8 13 33 fe ff 48 8b 44  |.D$.......3..H.D|
00044350  24 30 48 89 04 24 48 8b  44 24 20 48 89 44 24 08  |$0H..$H.D$ H.D$.|
00044350  24 30 48 89 04 24 48 8b  44 24 20 48 89 44 24 08  |$0H..$H.D$ H.D$.|
00044360  e8 fb 32 fe ff e8 36 2a  fe ff 8b 44 24 14 a9 00  |..2...6*...D$...|
00044360  e8 fb 32 fe ff e8 36 2a  fe ff 8b 44 24 14 a9 00  |..2...6*...D$...|
00044370  10 00 00 0f 85 bd 00 00  00 48 8b 44 24 18 48 83  |.........H.D$.H.|
00044370  10 00 00 0f 85 bd 00 00  00 48 8b 44 24 18 48 83  |.........H.D$.H.|
00044380  f8 01 7d 63 48 8b 44 24  48 48 8b 80 08 01 00 00  |..}cH.D$HH......|
00044380  f8 01 7d 63 48 8b 44 24  48 48 8b 80 08 01 00 00  |..}cH.D$HH......|
00044390  48 85 c0 75 2d e8 76 29  fe ff 48 8d 05 3c 0b 06  |H..u-.v)..H..<..|
00044390  48 85 c0 75 2d e8 76 29  fe ff 48 8d 05 3c 0b 06  |H..u-.v)..H..<..|
000443a0  00 48 89 04 24 48 c7 44  24 08 03 00 00 00 e8 ad  |.H..$H.D$.......|
000443a0  00 48 89 04 24 48 c7 44  24 08 03 00 00 00 e8 ad  |.H..$H.D$.......|
000443b0  32 fe ff e8 e8 29 fe ff  48 8b 6c 24 38 48 83 c4  |2....)..H.l$8H..|
000443b0  32 fe ff e8 e8 29 fe ff  48 8b 6c 24 38 48 83 c4  |2....)..H.l$8H..|
000443c0  40 c3 e8 49 29 fe ff 48  8d 05 12 22 06 00 48 89  |@..I)..H..."..H.|
000443c0  40 c3 e8 49 29 fe ff 48  8d 05 12 22 06 00 48 89  |@..I)..H..."..H.|
000443d0  04 24 48 c7 44 24 08 12  00 00 00 e8 80 32 fe ff  |.$H.D$.......2..|
000443d0  04 24 48 c7 44 24 08 12  00 00 00 e8 80 32 fe ff  |.$H.D$.......2..|
```

byte/machine code (ro)

Process (read only)



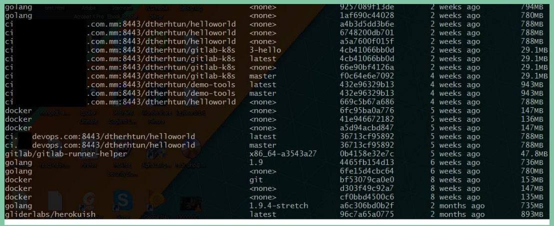
Docker



```
From golang:latest
MAINTAINER DTherHtun <dther@opslab.space>
ENV GOPATH /go
RUN go get -u github.com/gorilla/mux && mkdir /app
ADD . /app/
WORKDIR /app
RUN go build -o hw .
CMD ["./hw"]
~
~
~
~
```

Dockerfile

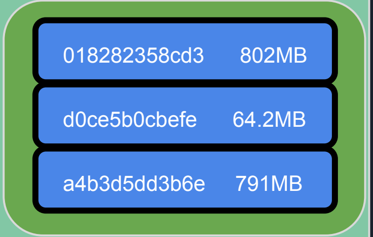
read only layer



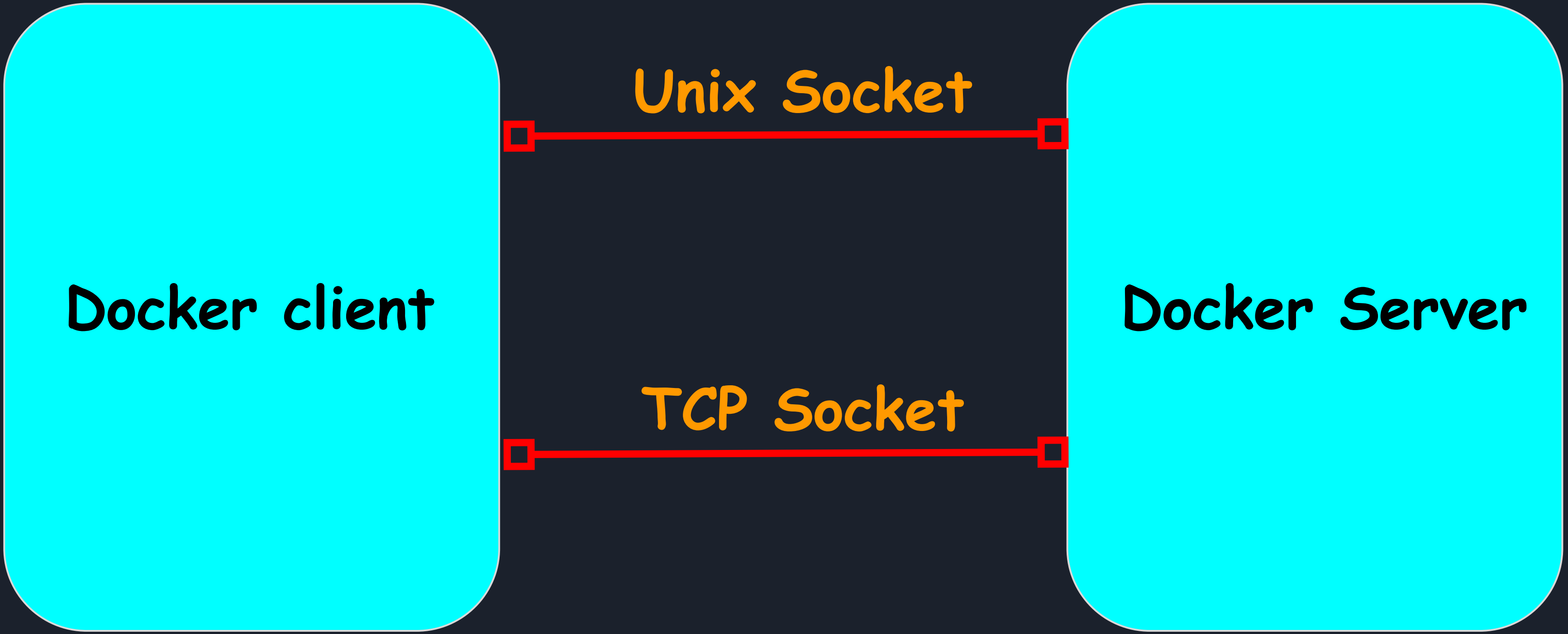
```
golang                                            <none>            9257089f13de    2 weeks ago     794MB
golang                                            <none>            1af690c44028    2 weeks ago     780MB
ci[redacted].com.mm:8443/dtherhtun/helloworld     <none>            a4b3d5dd3b6e    2 weeks ago     788MB
ci         .com.mm:8443/dtherhtun/helloworld      <none>            6748200db701    2 weeks ago     788MB
ci         .com.mm:8443/dtherhtun/helloworld      <none>            a5a7600f015f    2 weeks ago     788MB
ci         .com.mm:8443/dtherhtun/gitlab-k8s      3-hello           4cb41066bb0d    2 weeks ago     29.1MB
ci         .com.mm:8443/dtherhtun/gitlab-k8s      latest            4cb41066bb0d    2 weeks ago     29.1MB
ci         .com.mm:8443/dtherhtun/gitlab-k8s      <none>            66e90bf4126a    2 weeks ago     29.1MB
ci         .com.mm:8443/dtherhtun/gitlab-k8s      master            f0c64e6e7092    4 weeks ago     29.1MB
ci         .com.mm:8443/dtherhtun/demo-tools      latest            432e96329b13    4 weeks ago     943MB
ci         .com.mm:8443/dtherhtun/demo-tools      master            432e96329b13    4 weeks ago     943MB
ci         .com.mm:8443/dtherhtun/helloworld      <none>            669c5b67a686    4 weeks ago     788MB
docker                                            <none>            6fc95ba0a776    5 weeks ago     147MB
docker                                            <none>            41e946672182    5 weeks ago     136MB
docker                                            <none>            a5d94acbd847    5 weeks ago     147MB
ci.[redacted]devops.com:8443/dtherhtun/helloworld  latest            36713cf95892    5 weeks ago     788MB
ci.[redacted]devops.com:8443/dtherhtun/helloworld  master            36713cf95892    5 weeks ago     788MB
gitlab/gitlab-runner-helper                       x86_64-a3543a27   0b4158e32e7c    5 weeks ago     47.8MB
golang                                            1.9               4465fb154d13    6 weeks ago     736MB
golang                                            <none>            6fe15d4cbc64    6 weeks ago     780MB
docker                                            git               bf53079ca0e0    8 weeks ago     153MB
docker                                            <none>            d303f49c92a7    8 weeks ago     147MB
docker                                            <none>            cf0bbd4500c6    8 weeks ago     135MB
golang                                            1.9.4-stretch     a6c306bd0b2f    2 months ago    735MB
gliderlabs/herokuish                              latest            96c7a65a0775    2 months ago    893MB
```

Images

read onlay layer + write layer



Docker container


Docker client
Unix Socket
TCP Socket
Docker Server

Docker Installing

- CentOs
  ```
  $ sudo yum install docker -y
  ```

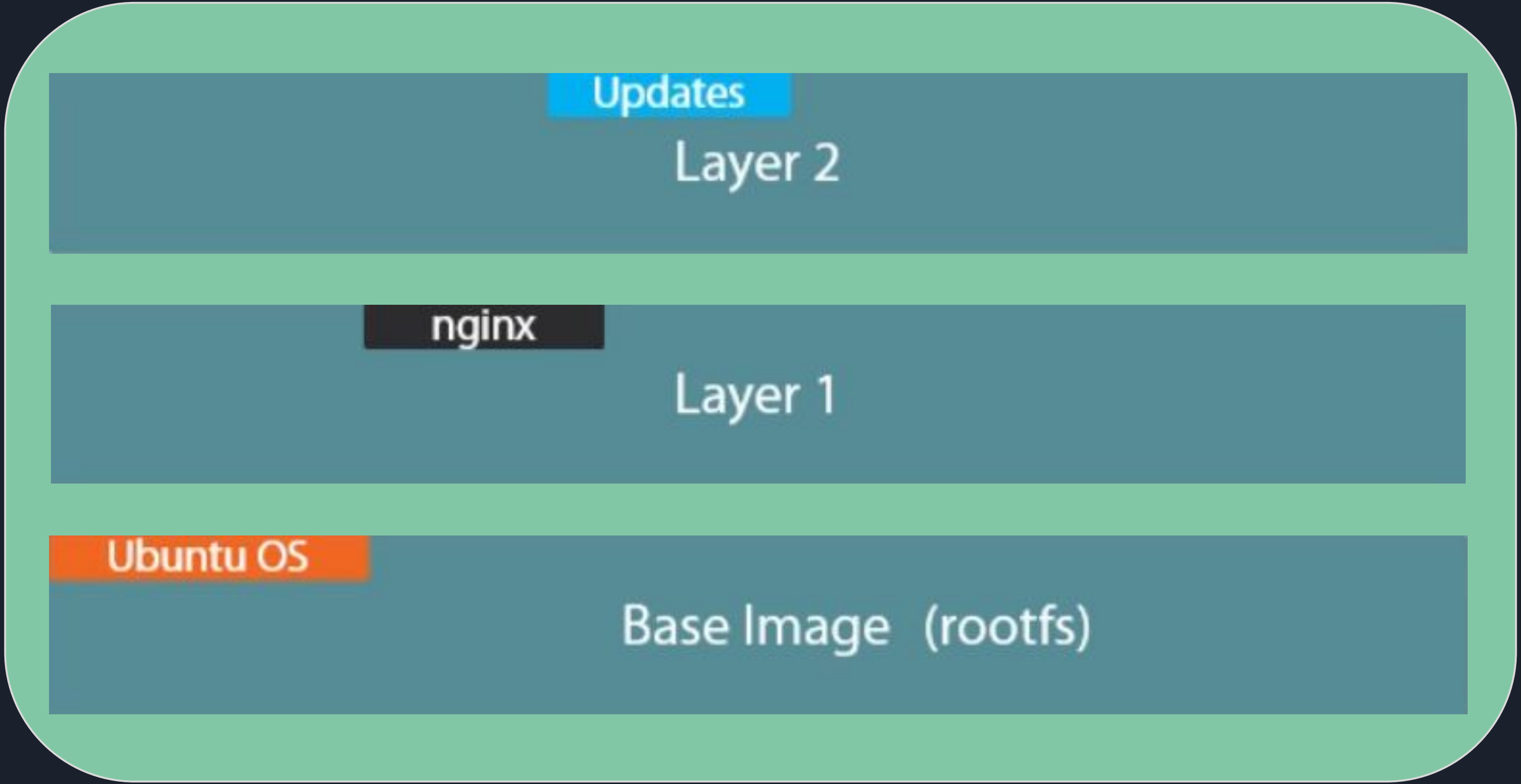
- Ubuntu
  ```
  $ sudo apt-get update -y && sudo apt-get install docker -y
  ```

- Granting Docker control to non-root users
  ```
  $ sudo gpasswd -a user docker
  ```

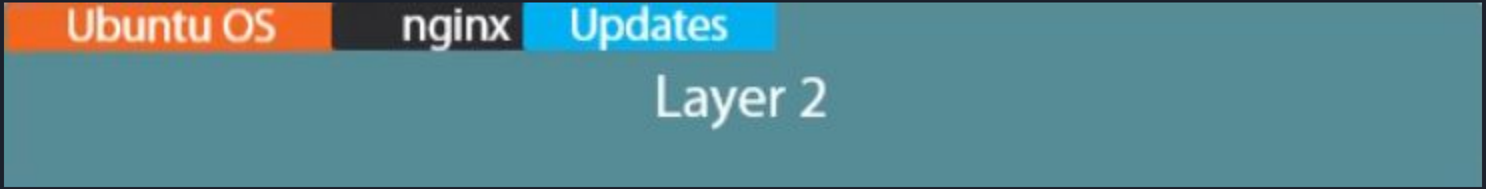
- Configuring Docker to communicate over the network
  - Docker Daemon
    ```
    $ sudo dockerd -H 0.0.0.0:2375 &&
    ```
    Or
    ```
    $ sudo vim /etc/sysconfig/docker
      OPTIONS='--selinux-enabled -H=unix:///var/run/docker.sock -H=0.0.0.0:2375'
    ```

  - Docker client
    ```
    $ export DOCKER_HOST="tcp://ip:port"
    ```

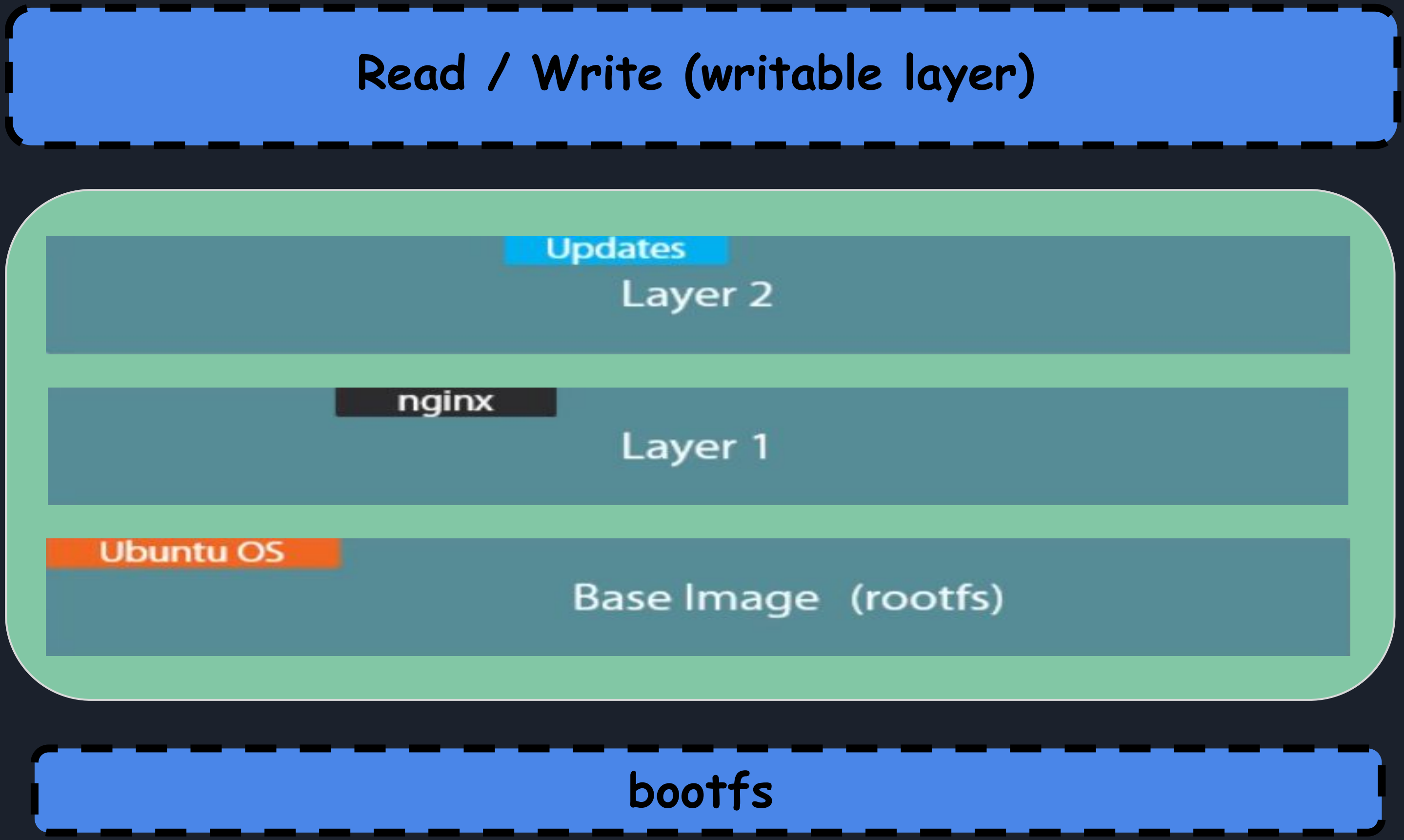
Docker Images Layers



Docker Image Layers



Magic of union mounts



When we launch a container

Docker Image Layers Gist

- ★ Combined into a single view of all layers. The look feel of a single regular everyday file system.
- ★ The higher layers hiding the data in the lower layers.
- ★ When conflict, higher layer every win.
- ★ Union mounts is all of these layers in the image are mounted as read-only and the top layer, additional layer which is added when we launch a container, is the only writable layer.
- ★ Actually a small bootfs that exists below the rootfs. It's not there after the container is started.
- ★ All changes to the container at runtime are committed to this top layer via copy on write behavior.

A little Play with Docker engine and container

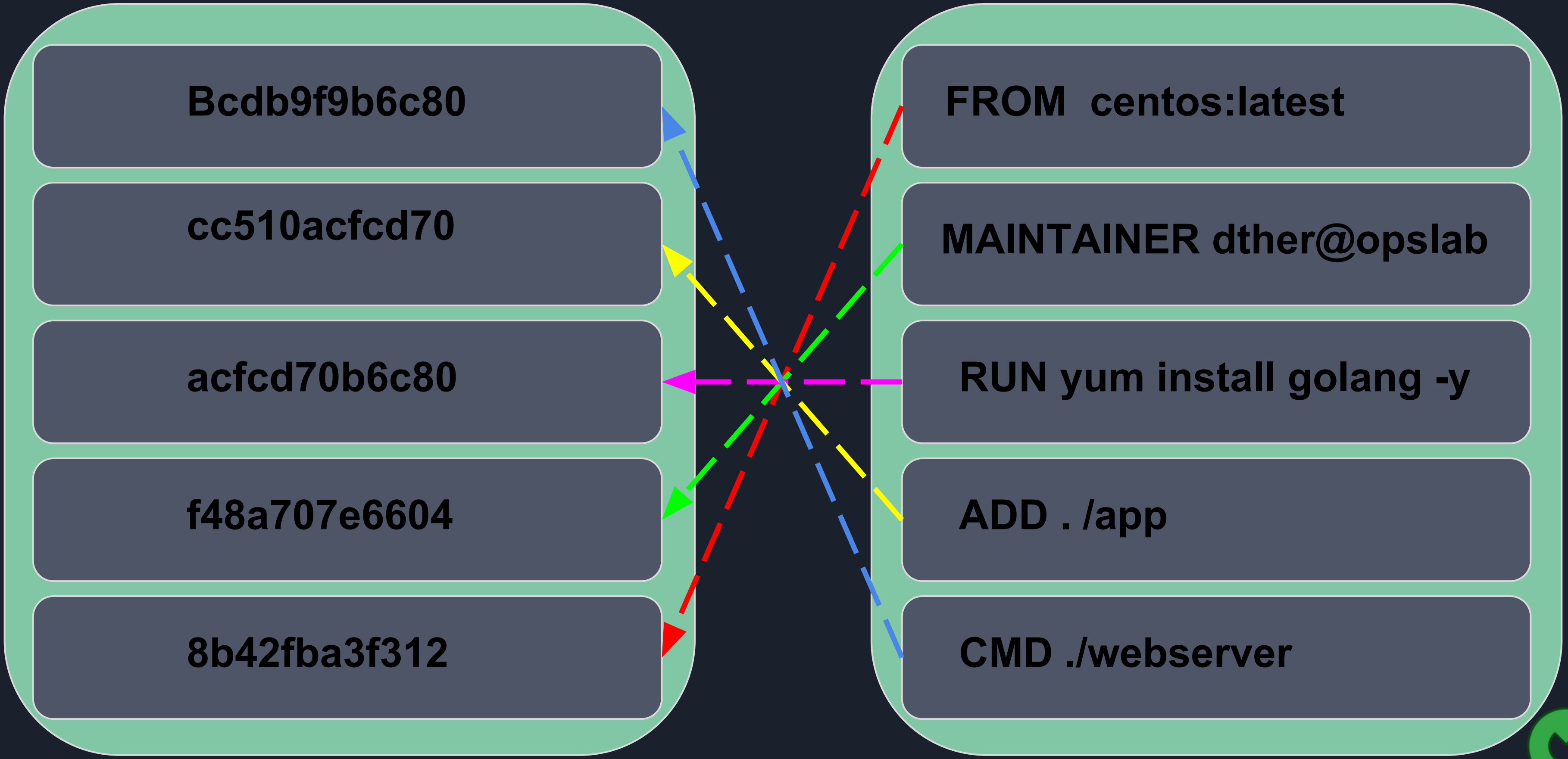
A little play with docker engine and containers

- Copy Images to another host
  - docker commit container-id name
  - docker save -o /tmp/something.tar image
  - docker load -i /path/to/something.tar
- One Process per Container
  - Docker runs one process per container by default
  - The UNIX philosophy “Do One Thing Well”
  - Good for web scaler apps and microservices architecture and the likes.
- Container management
  - Run 'docker COMMAND --help' for more information on a command.
  - https://docs.docker.com/engine/reference/run/
- Getting a shell in a container
  - Attaches to PID 1 inside the container. (But In the realworld, PID 1 inside a container will probably not be a shell.)
  - ssh (will not be running ssh service mostly containers.)
  - nsenter (Enter into Namespace) . requires the containers PID
  - Docker exec -it container /bin/bash

Docker Build

Docker Images Layers

Dockerfile



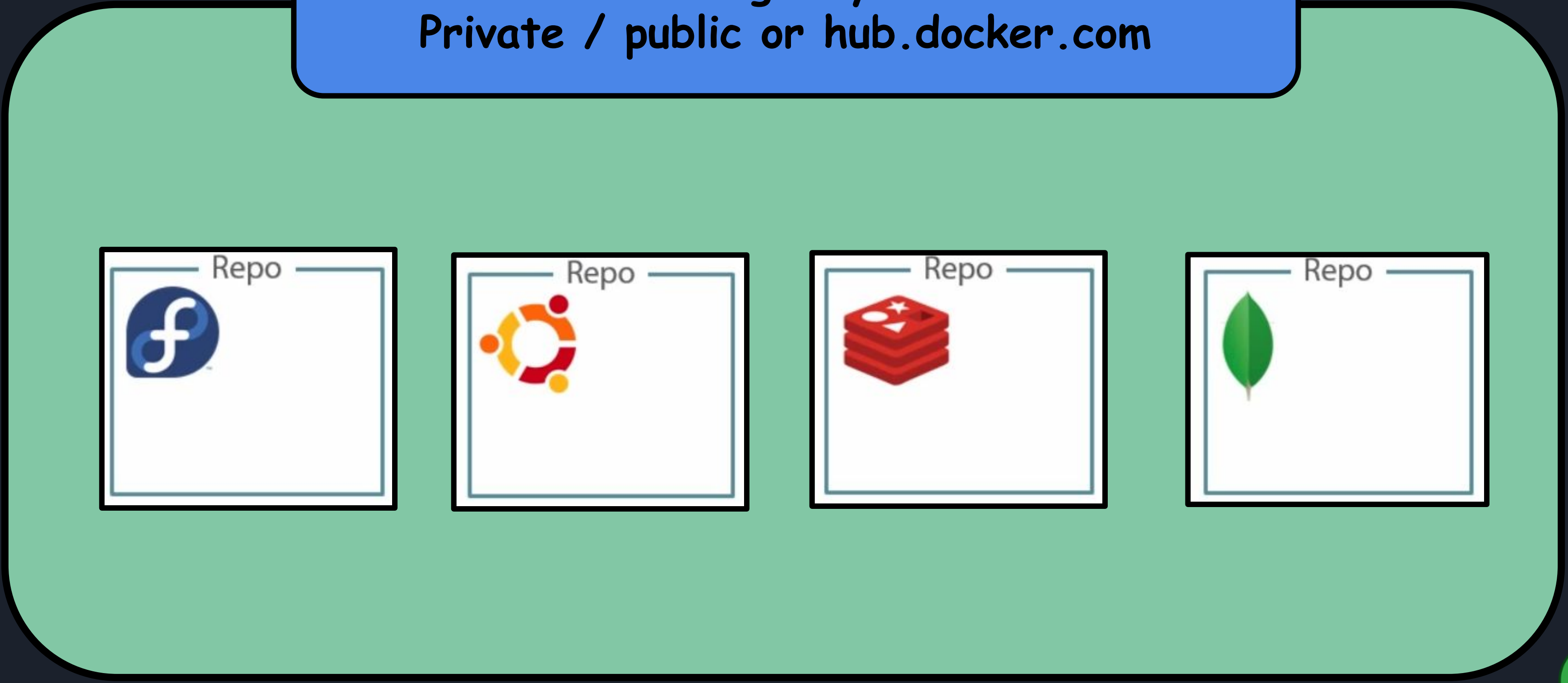
```
FROM  centos:latest
MAINTAINER dther@opslab
RUN yum install golang -y
ADD . /app
CMD ./webserver
```

Docker Build Gist

- What is Dockerfile?
  - Plain text
  - Simple format
  - Instructions to build docker image
- Docker Instructions
  - FROM (set the base image)
  - MAINTAINER (indicate to author/ anywhere in Dockerfile/ just metadata)
  - RUN (every RUN instruction create a new layer in image)
  - CMD (run command / doesn't execute anything during the build time.)
  - ENTRYPOINT (primary command for image)
  - COPY (src -> dest / in the file system of the container)
  - ADD (similar COPY but, external container)
  - VOLUME (enable access to a location on the host system from container)
  - WORKDIR (set the currently active directory for other instructions)
  - ENV (Environment Variable key=value)
  - Detail at following link - https://deis.com/blog/2015/dockerfile-instructions-syntax/

Docker Registry

Registry
Private / public or hub.docker.com



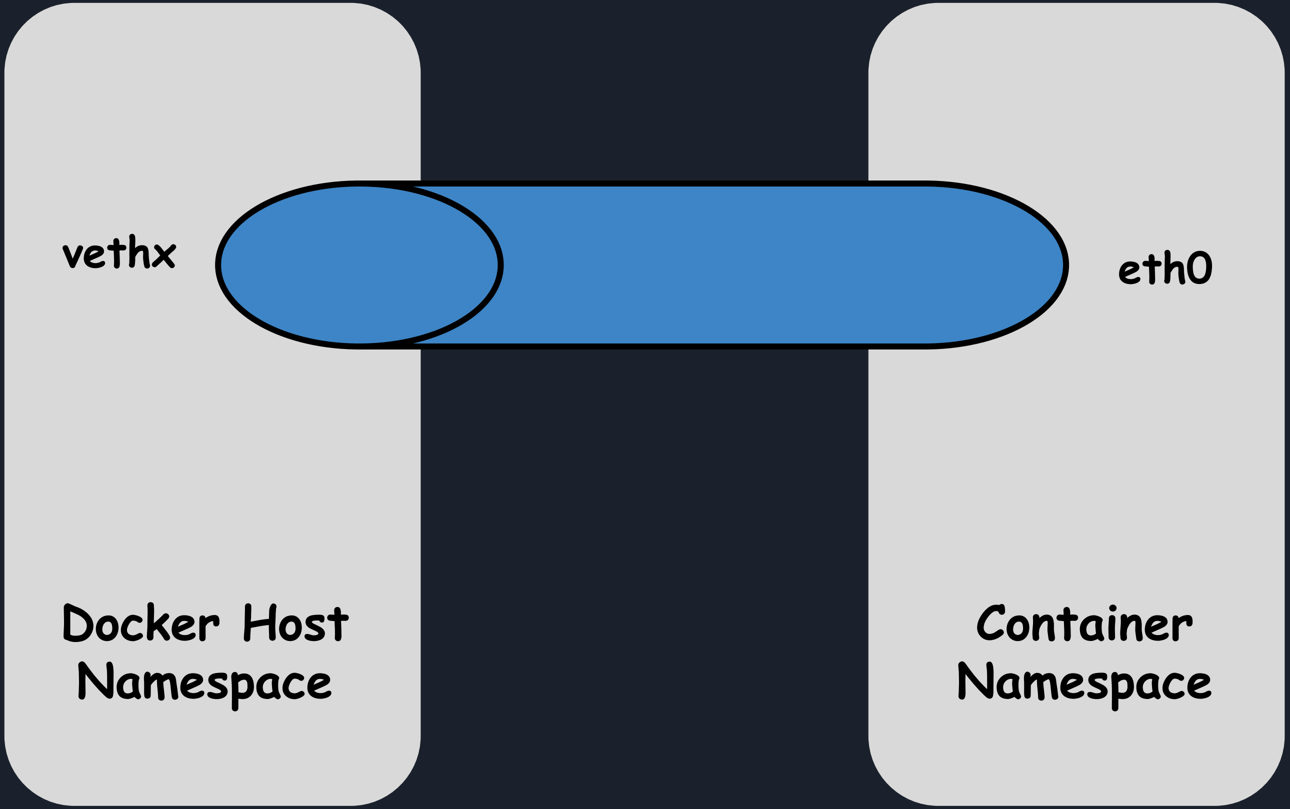
Docker Registry Gist

- Cloud Registry
  - hub.docker.com (public/private)
  - quay.io (public/private)
  - coreos.com/quay-enterprise (private)
  - bintray.com (private)
- Local Registry
  - `$ docker run -d --restart=always -p 5000:5000 --name registry registry`

  - Usage
    - `$ vim /etc/docker/daemon.json`

      ```
      { "insecure-registries":["myregistry.example.com:5000"] }
      ```
    - `$ systemctl restart docker`
    - `$ docker tag container-id [REGISTRYHOST/][USERNAME/]NAME[:TAG]`
    - `$ docker push NAME[:TAG]`

Docker Networking



Theory is pretty much like a pipe or a tube

Docker Networking Gist

- ❏ Docker daemon start on our host and it create docker0 virtual bridge.
- ❏ By default each new container get one interface.
- ❏ It automatically attached to the docker0 virtual bridge.
- ❏ Linking container is more secure than exposing ports
- ❏ Linking container is only works for container to container communication

```
$ brctl show docker0
$ docker run --dns=1.1.1.1 --name=dnstest busybox
$ docker run -d -p hostport:containerport:containerport --name=web2 mosc-web
$ docker port mosc-web
$ docker run --name=rcvr --link=src:ali-src -it centos /bin/bash
```

Docker Troubleshoot

Docker Troubleshoot Gist

- Docker daemon logging mode
  - debug, info, error, fatal
  - fatal only logs fatal messages
  - error logs error and fatal
  - info logs infos, error, fatal
  - debug, logs them all
  - $ vim /etc/sysconfig/docker

    ```
    "--log-level=fatal"
    ```
- Docker Image Troubleshooting
  - Test Before Dockerfile
- Docker Network Troubleshooting
  - very basic network checking
  - which range of ip addresses to assign to the docker0 bridge
  - $ nmcli connection down docker0
  - $ vim /etc/sysconfig/docker

    ```
    " --bip=150.150.0.1/24"
    ```
  - Inter container communication (--icc= / --iptables)
  - default value of both setting is true

--=END=--