SAY HI TO VIRTUALIZATION

DOCKER

Wang Xuechen

2019/09/02

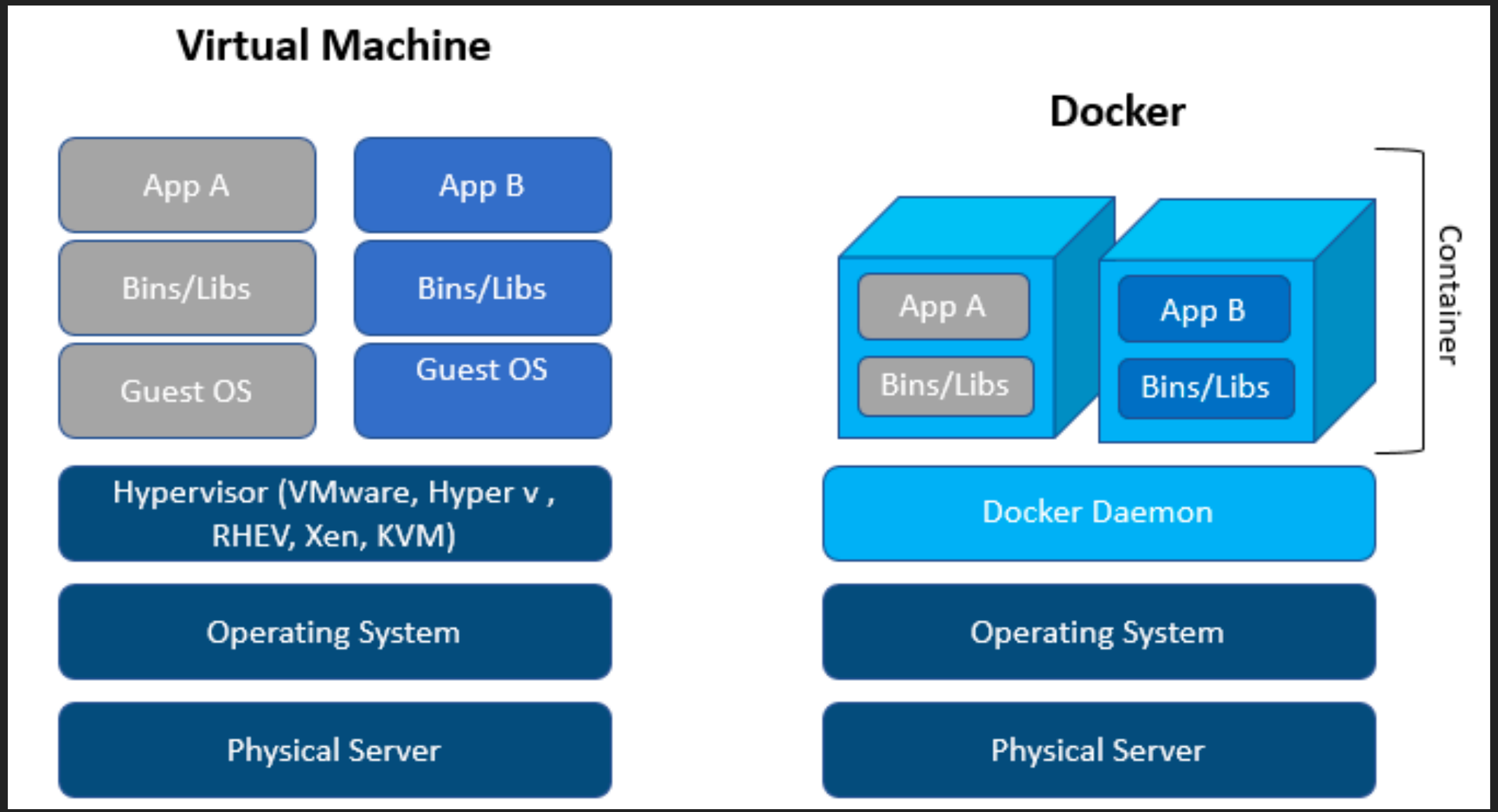
AGENDA

- What's Docker?
  - Different from other virtualization technologies
  - Architecture
  - Networking Model
  - General workflow and Using occasions
- Docker Hub

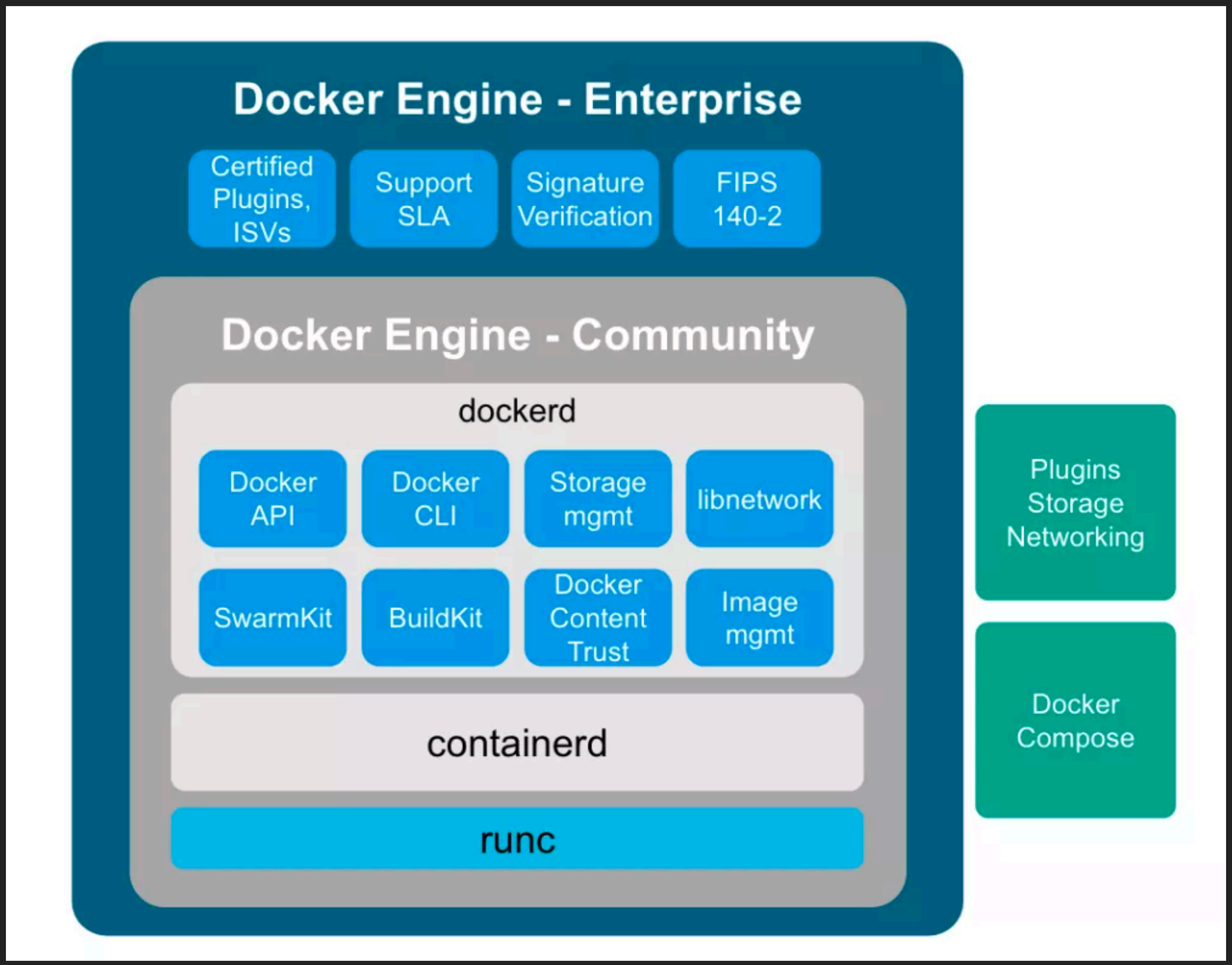
AGENDA

- How to
  - Install Docker-CE
  - Docker CLI (docker) / Dockerfile
  - Docker Compose File & CLI
  - Docker Swarm*

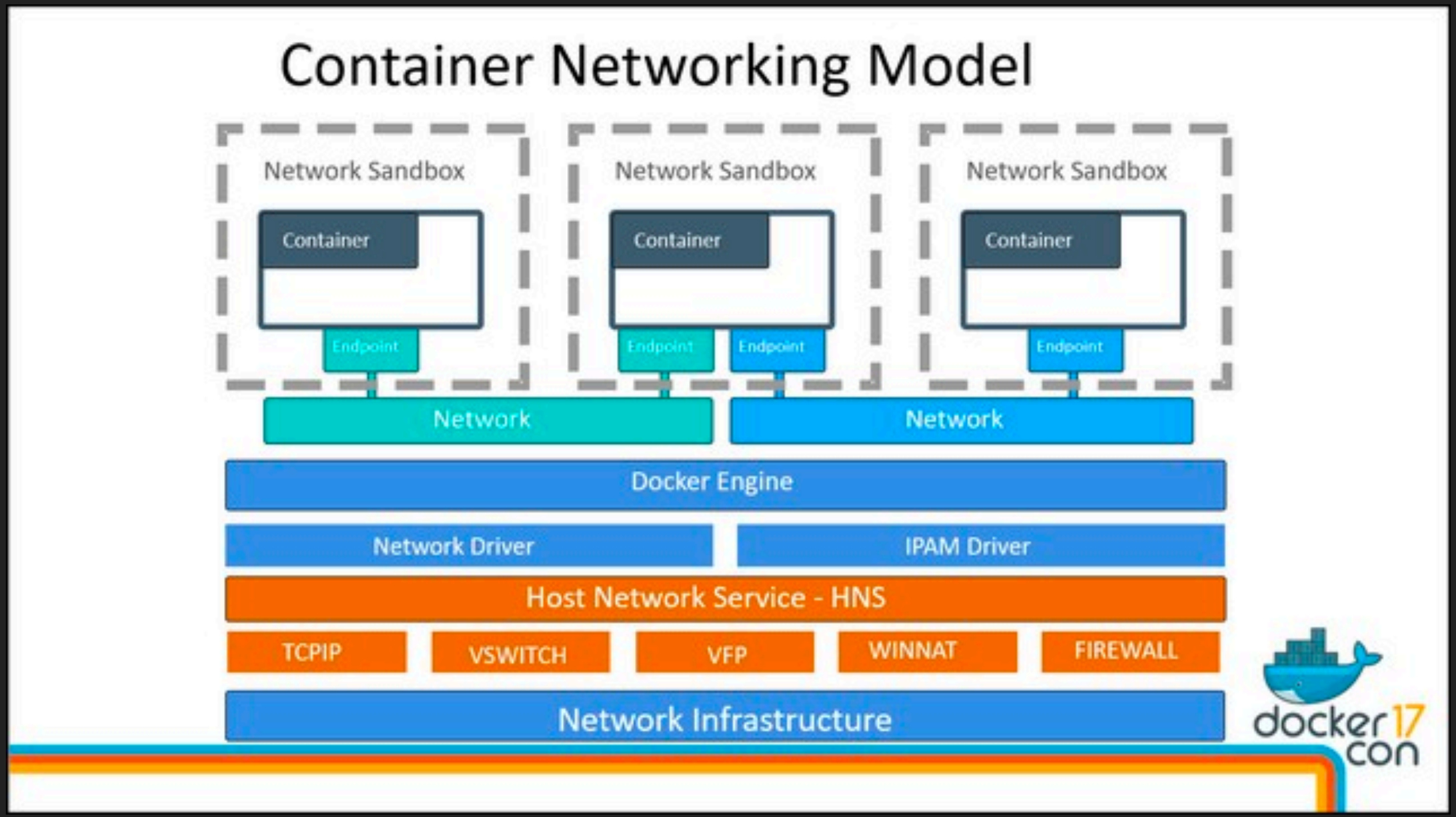
DIFFERENT


Virtual Machine
App A
App B
Bins/Libs
Bins/Libs
Guest OS
Guest OS
Hypervisor (VMware, Hyper v , RHEV, Xen, KVM)
Operating System
Physical Server
Docker
App A
App B
Bins/Libs
Bins/Libs
Container
Docker Daemon
Operating System
Physical Server

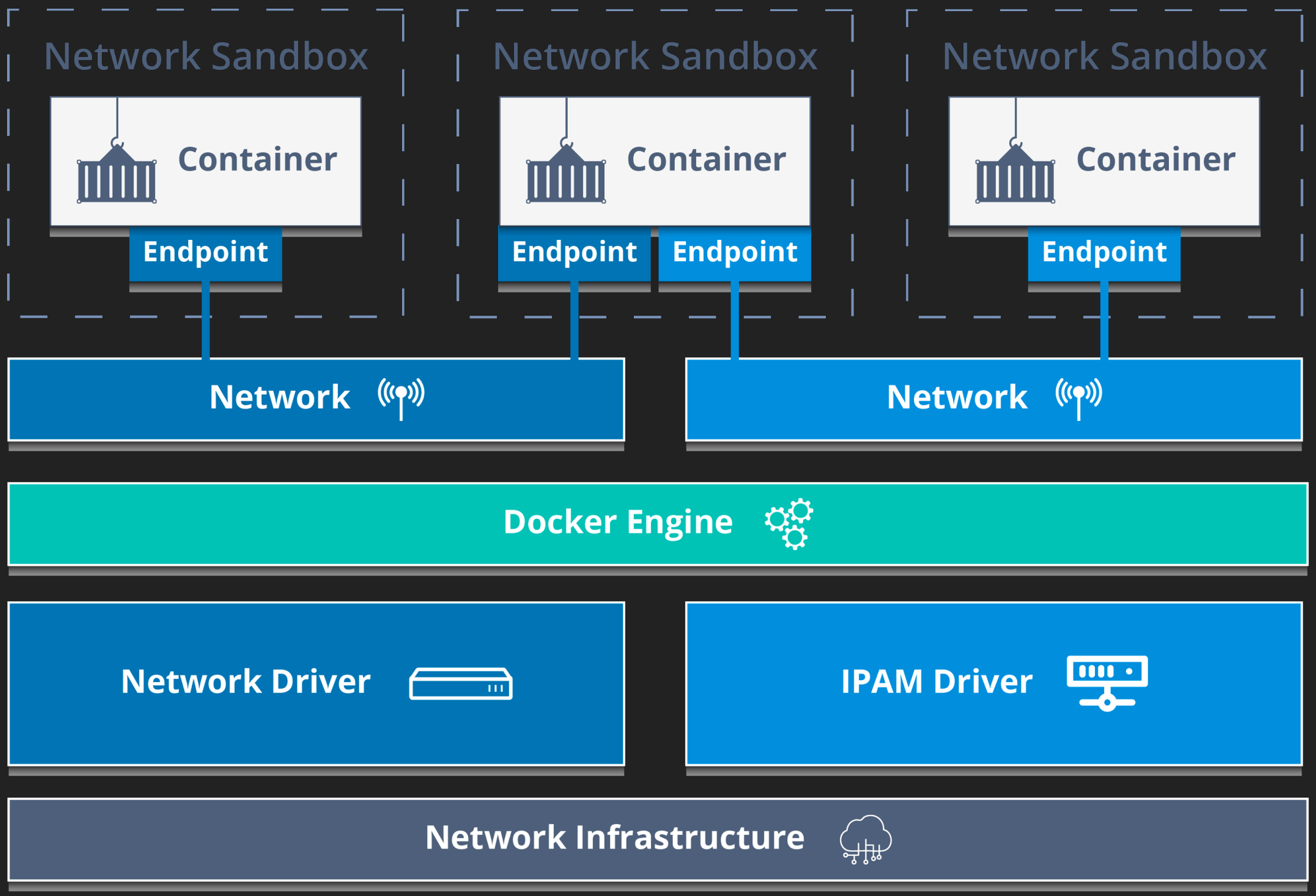
ARCHITECTURE



DOCKER NETWORKING

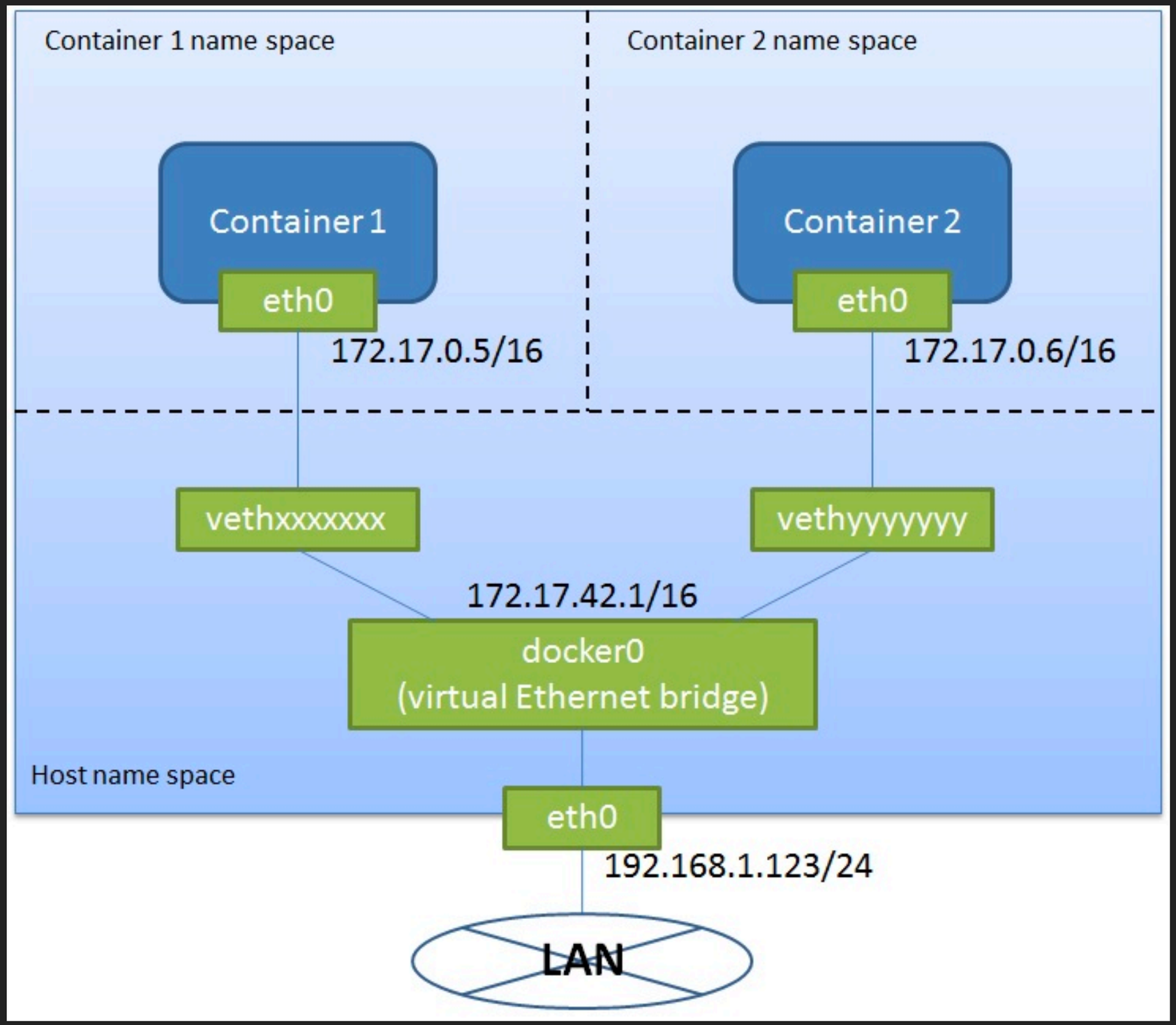


DOCKER NETWORKING

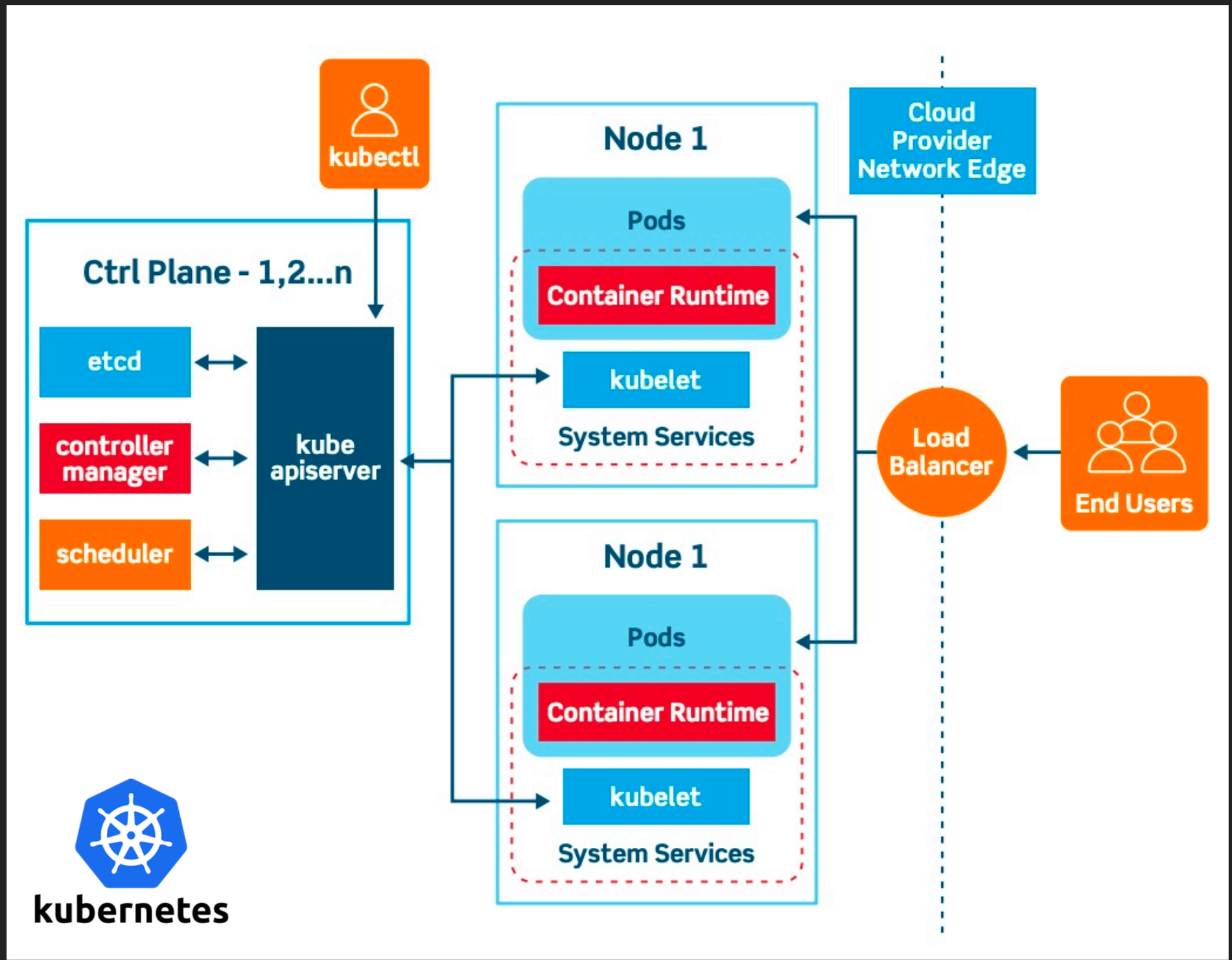


Architecture of Container Networking Model

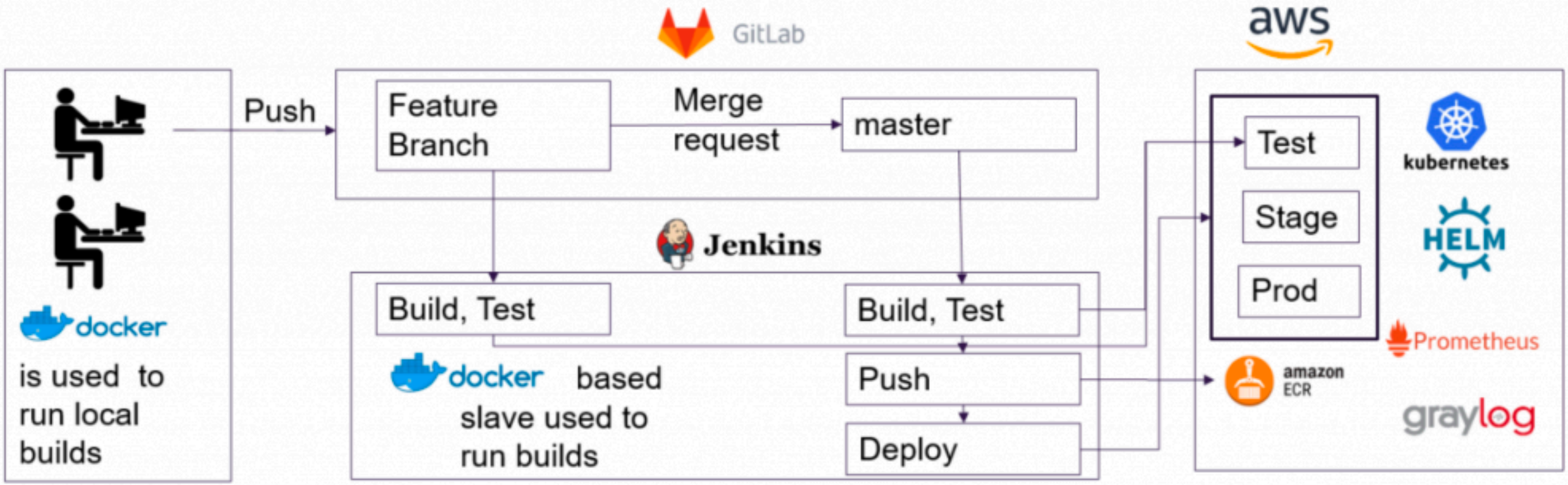
DOCKER NETWORKING



K8S


kubectl
Ctrl Plane - 1,2...n
etcd
controller manager
scheduler
kube apiserver
Node 1
Pods
Container Runtime
kubelet
System Services
Node 1
Pods
Container Runtime
kubelet
System Services
Cloud Provider Network Edge
Load Balancer
End Users
kubernetes

WORKFLOW



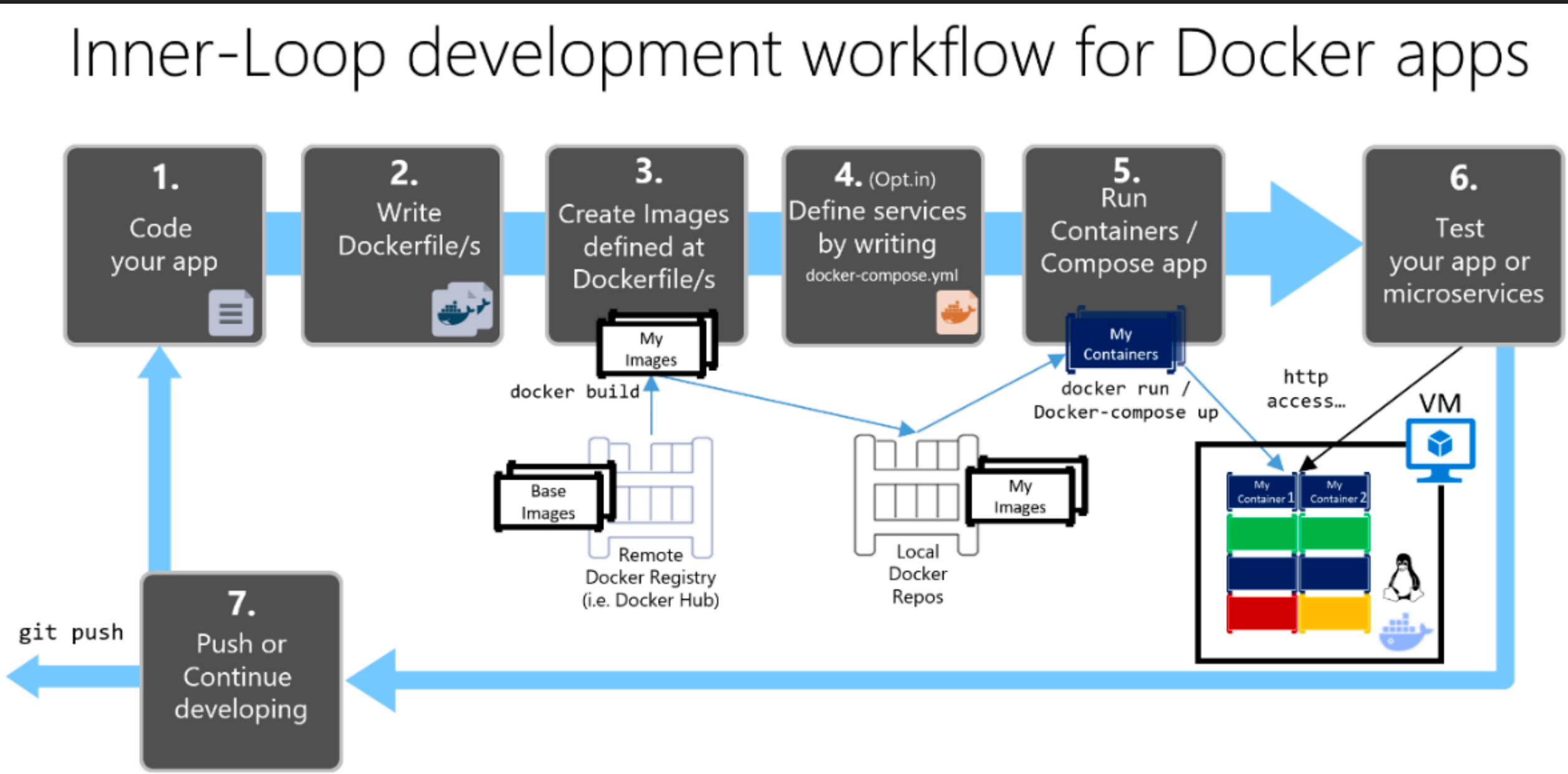
Local
- IDE - develop
- Docker - build image and test

Feature Branch
- Jenkins - run build script
- Docker – build image and test
- System test – using Test environment

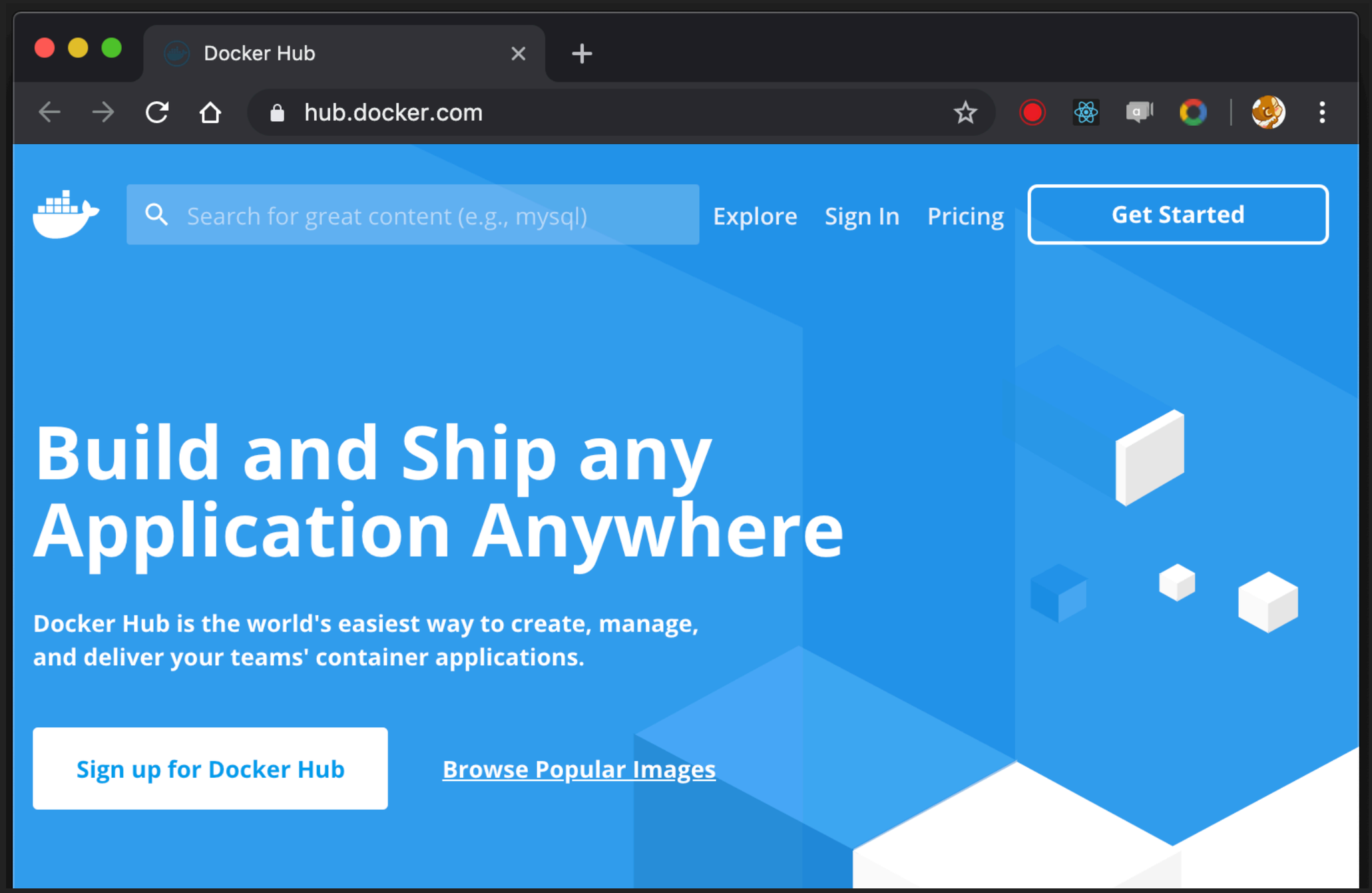
Master Branch
- Same as feature branch
- Deployment

Runtime
- ECR – Docker registry
- Kubernetes - manage containers
- Gray log- Log aggregation
- Prometheus – collect counters

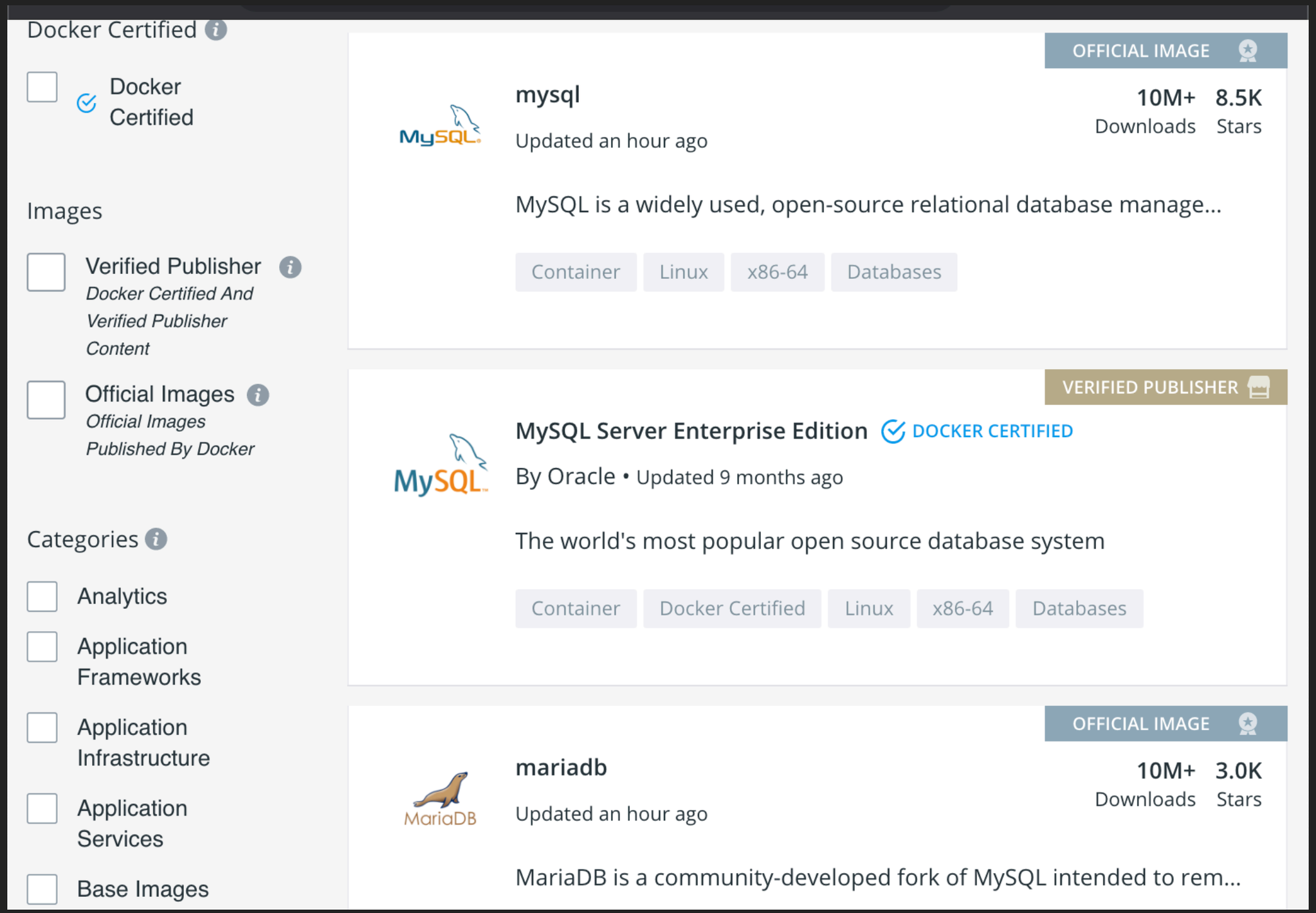
WORKFLOW


Inner-Loop development workflow for Docker apps
1. Code your app
2. Write Dockerfile/s
3. Create Images defined at Dockerfile/s
4. (Opt.in) Define services by writing docker-compose.yml
5. Run Containers / Compose app
6. Test your app or microservices
My Images
docker build
Base Images
Remote Docker Registry (i.e. Docker Hub)
Local Docker Repos
My Images
My Containers
docker run / Docker-compose up
http access…
VM
My Container 1
My Container 2
git push
7. Push or Continue developing

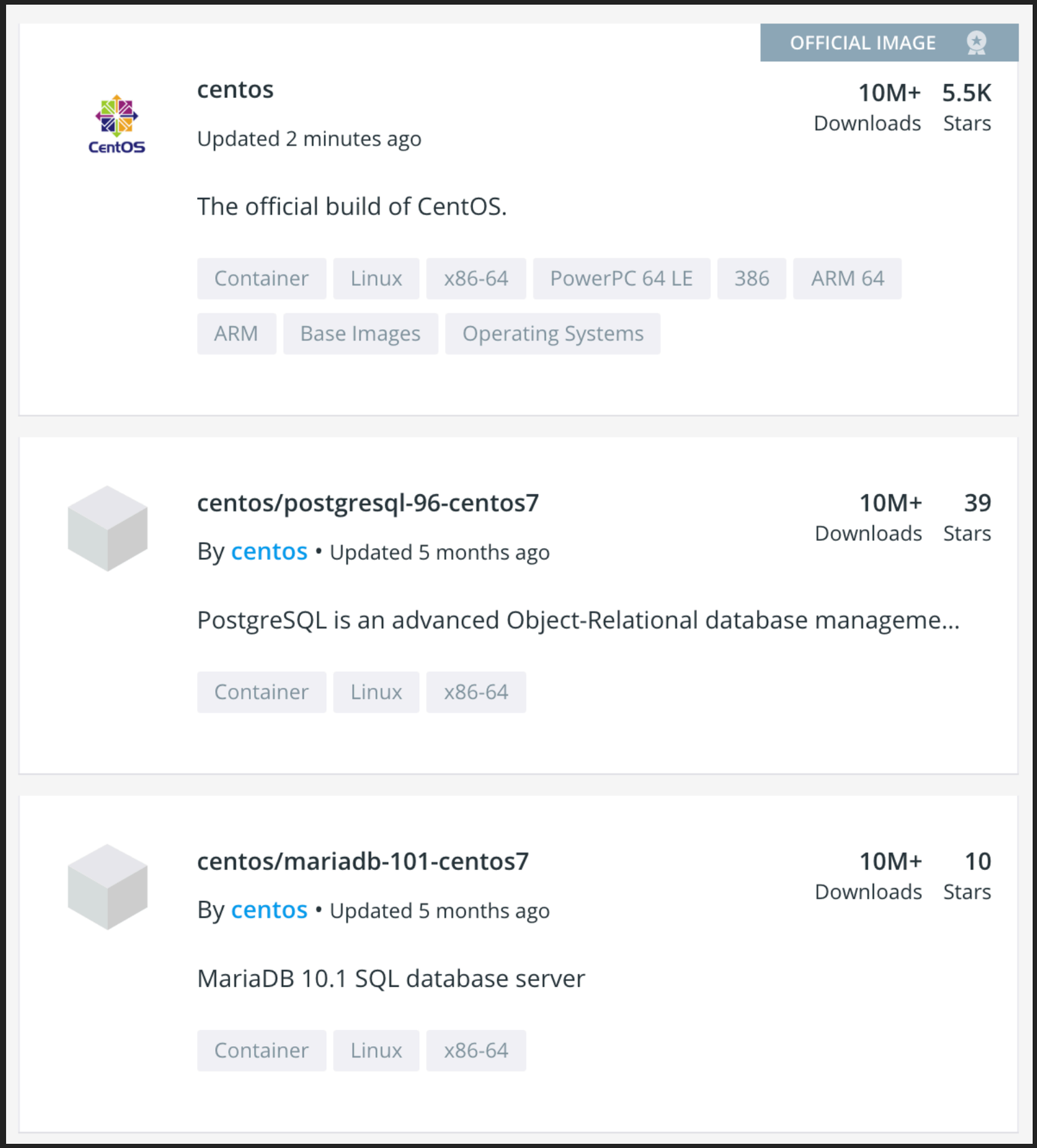
DOCKER HUB


Docker Hub
hub.docker.com
Search for great content (e.g., mysql)
Explore
Sign In
Pricing
Get Started
Build and Ship any Application Anywhere
Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.
Sign up for Docker Hub
Browse Popular Images

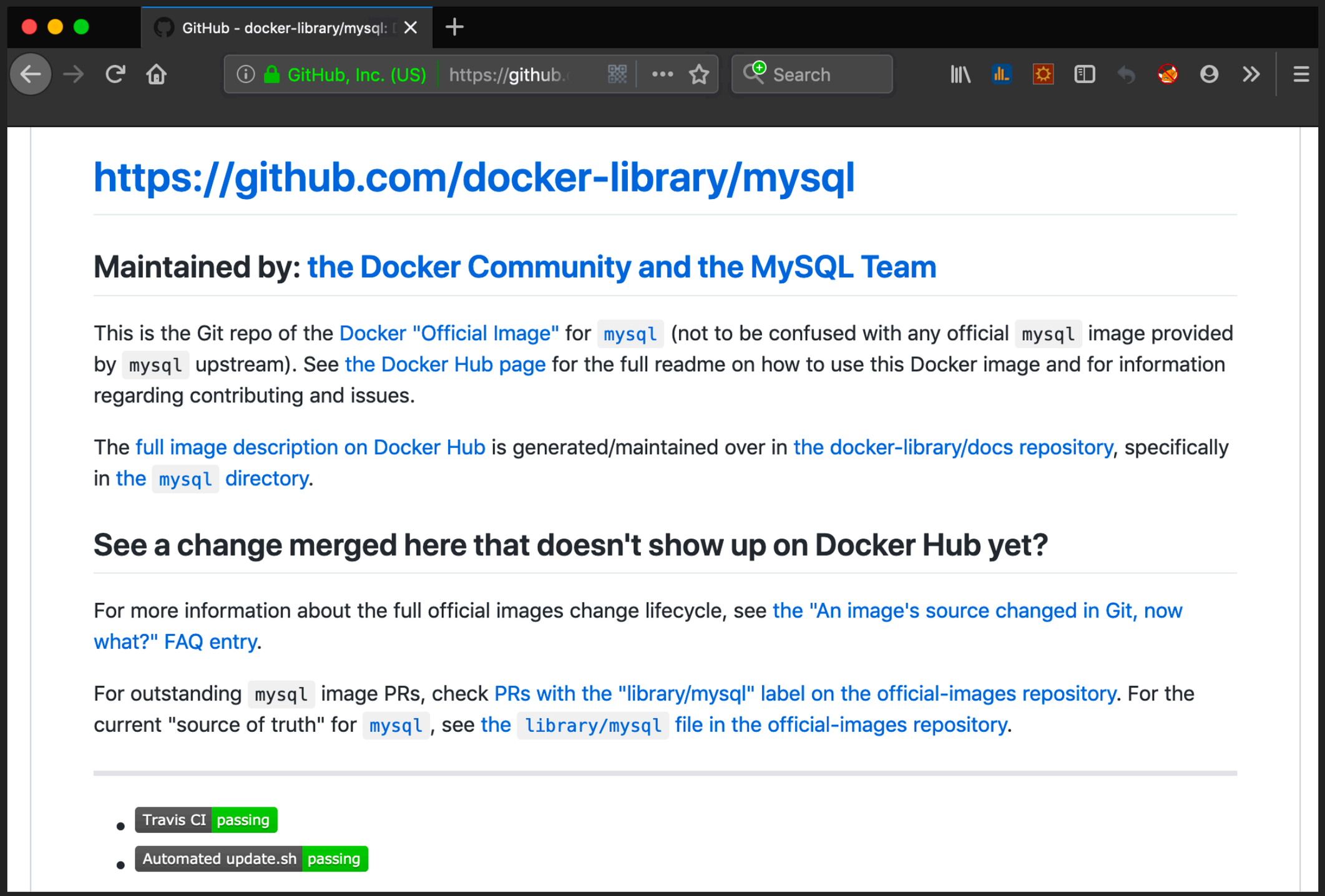
DOCKER HUB


Docker Certified
Docker Certified
Images
Verified Publisher
Docker Certified And Verified Publisher Content
Official Images
Official Images Published By Docker
Categories
Analytics
Application Frameworks
Application Infrastructure
Application Services
Base Images
OFFICIAL IMAGE
mysql
Updated an hour ago
10M+ Downloads
8.5K Stars
MySQL is a widely used, open-source relational database manage...
Container
Linux
x86-64
Databases
VERIFIED PUBLISHER
MySQL Server Enterprise Edition
DOCKER CERTIFIED
By Oracle • Updated 9 months ago
The world's most popular open source database system
Container
Docker Certified
Linux
x86-64
Databases
OFFICIAL IMAGE
mariadb
Updated an hour ago
10M+ Downloads
3.0K Stars
MariaDB is a community-developed fork of MySQL intended to rem...

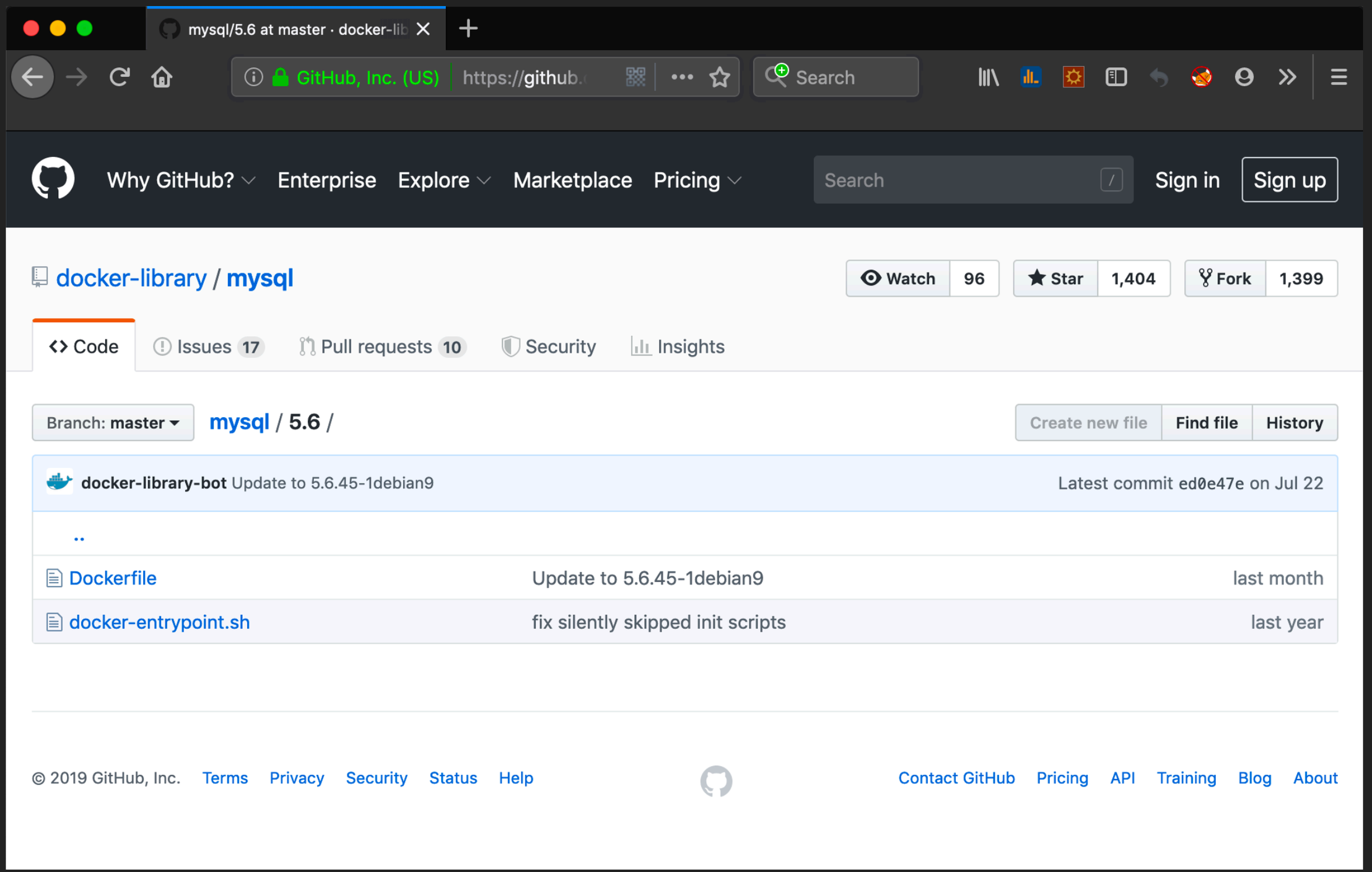
DOCKER HUB


OFFICIAL IMAGE
centos
Updated 2 minutes ago
10M+ Downloads
5.5K Stars
The official build of CentOS.
Container
Linux
x86-64
PowerPC 64 LE
386
ARM 64
ARM
Base Images
Operating Systems
centos/postgresql-96-centos7
By centos • Updated 5 months ago
10M+ Downloads
39 Stars
PostgreSQL is an advanced Object-Relational database manageme...
Container
Linux
x86-64
centos/mariadb-101-centos7
By centos • Updated 5 months ago
10M+ Downloads
10 Stars
MariaDB 10.1 SQL database server
Container
Linux
x86-64

DOCKER HUB & GIT HUB



DOCKER HUB & GIT HUB



ECOSYSTEM

Service Providers

rackspace the open cloud company · DEIS · Google Cloud Platform · amazon web services · IBM · SOFTLAYER an IBM Company · openstack · Yandex · OPENSHIFT · Microsoft Azure · DigitalOcean · CLOUD FOUNDRY · linode · Quay.io · Baidu 百度 · Baremetal Industries · clever cloud · Google Compute Engine · stackato by ActiveState · Google App Engine · domino · StackDock · Gilliam · tutum · ORCHARD · Flynn · VOXOZ

Operating Systems

redhat · boot2docker · CoreOS · ubuntu · debian · fedora · CentOS

Configuration Management

puppet labs · Chef · Jenkins · Capistrano · DotCi · Packer · ANSIBLE · SALTSTACK

Dev Tools

codefresh · Fig · VAGRANT · heroku · Shippable · runnable · NODECHECKER · wercker · drone.io

Big Data

OpenCore · syncsort · hadoop

Service Discovery

NETFLIX Eureka · etcd · SkyDNS

Official Repositories

mongoDB · MySQL · PostgreSQL · Ruby · WordPress · Java · debian · CentOS · ubuntu · RAILS · redis · NGINX · node

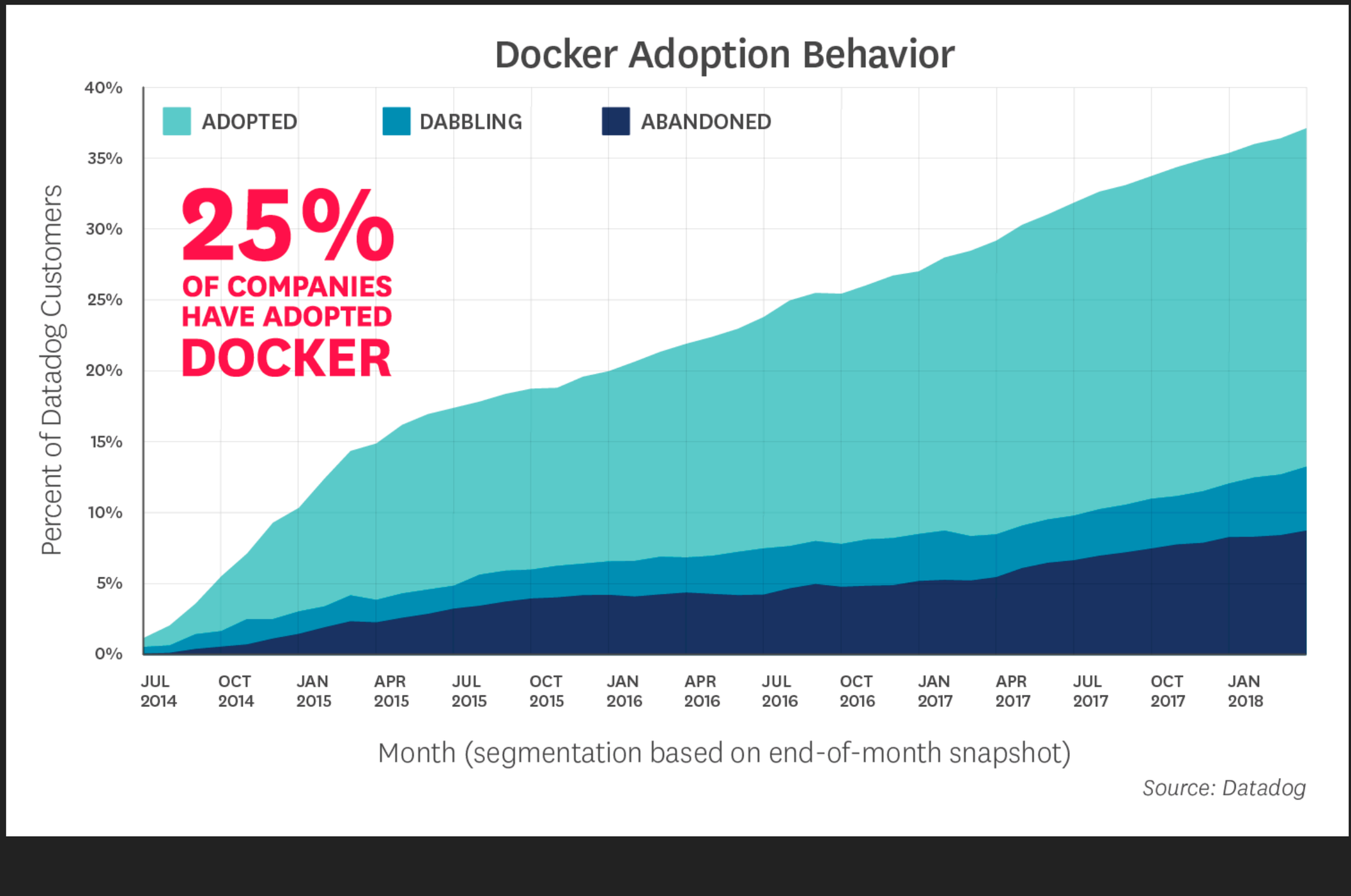
Orchestration

MESOS · Shipper · CONSUL · VNS3 · Marathon · fleet · ATOMIC

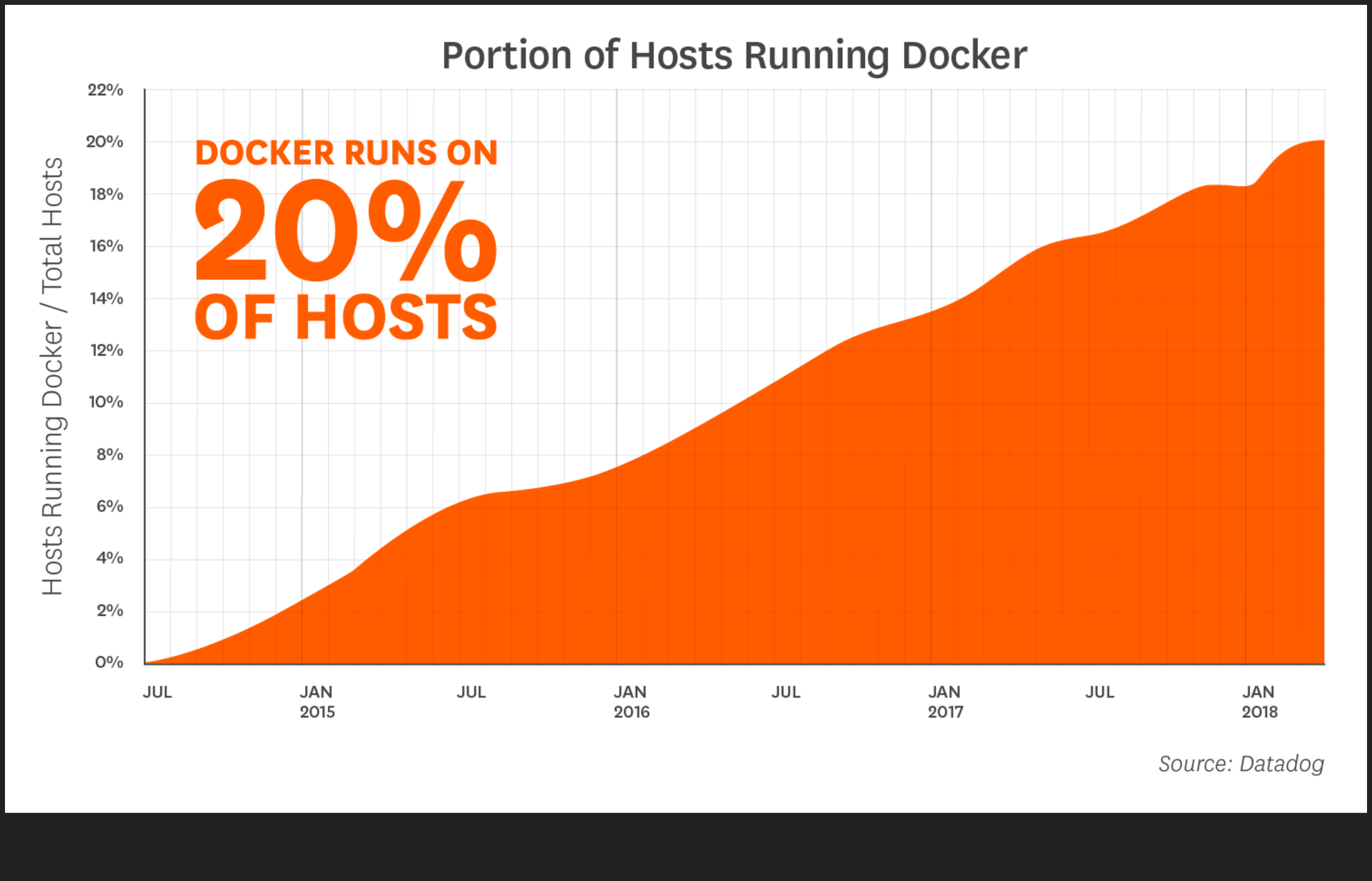
System Integrators

OpDemand · Wiredcraft · MomentumSI · InfoSiftr · Flux7 · SHADOW SOFT · Dev9 · relevance lab · IZURI

DOCKER-ADOPTION @ DATADOGHQ.COM



DOCKER-ADOPTION @ DATADOGHQ.COM

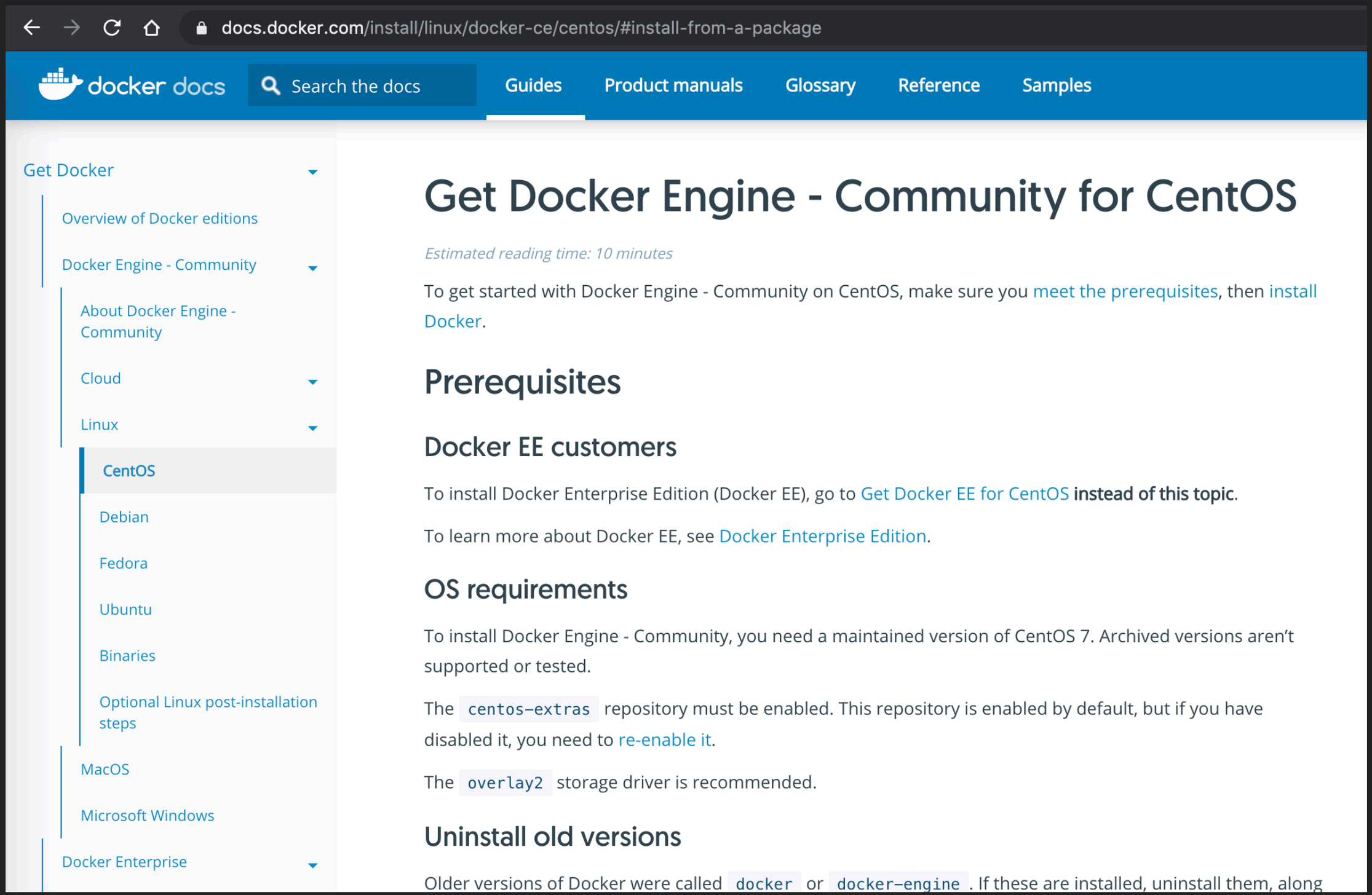


Source: Datadog

DOCKER-ADOPTION @ DATADOGHQ.COM

- https://www.datadoghq.com/docker-adoption/

INSTALL DOCKER-CE



INSTALL DOCKER-CE

docs.docker.com/install/linux/docker-ce/centos/

docker docs | Guides | Product manuals | Glossary | Reference | Samples

- Get Docker
  - Overview of Docker editions
  - Docker Engine - Community
    - About Docker Engine - Community
  - Cloud
  - Linux
    - CentOS
    - Debian
    - Fedora
    - Ubuntu
    - Binaries
    - Optional Linux post-installation steps
  - MacOS
  - Microsoft Windows

Install from a package

If you cannot use Docker's repository to install Docker, you can download the `.rpm` file for your release and install it manually. You need to download a new file each time you want to upgrade Docker Engine - Community.

1. Go to https://download.docker.com/linux/centos/7/x86_64/stable/Packages/ and download the `.rpm` file for the Docker version you want to install.

docs.docker.com/install/linux/docker-ce/centos/

docker docs | Search the docs | Guides | Product manuals | Glossary | Reference | Samples

- Get Docker
  - Overview of Docker editions
  - Docker Engine - Community
    - About Docker Engine - Community
  - Cloud
  - Linux
    - CentOS
    - Debian
    - Fedora
    - Ubuntu
    - Binaries

Install using the repository

Before you install Docker Engine - Community for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

SET UP THE REPOSITORY

1. Install required packages. `yum-utils` provides the `yum-config-manager` utility, and `device-mapper-persistent-data` and `lvm2` are required by the `devicemapper` storage driver.

```
$ sudo yum install -y yum-utils \
  device-mapper-persistent-data \
  lvm2
```

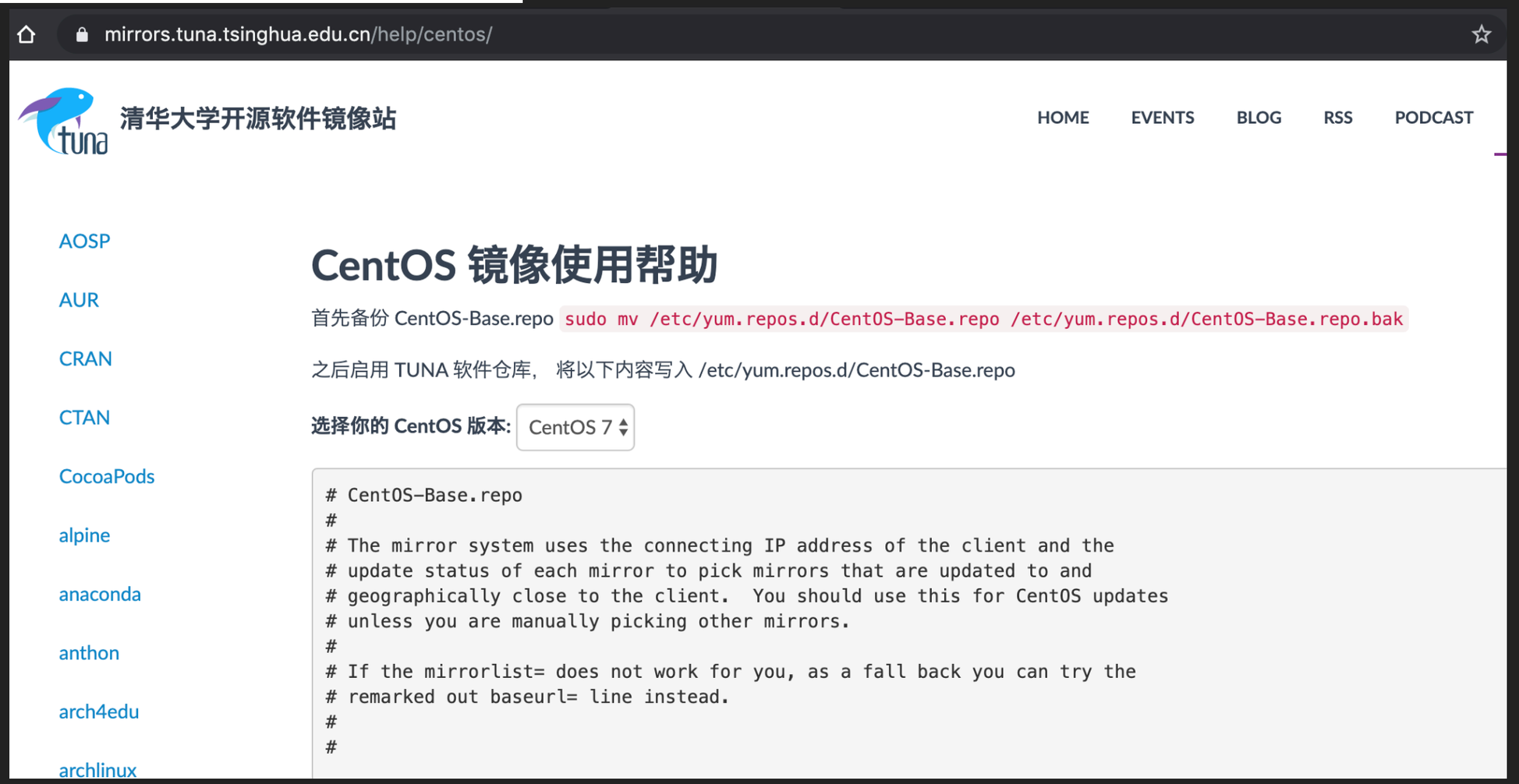
2. Use the following command to set up the **stable** repository.

```
$ sudo yum-config-manager \
    --add-repo \
    https://download.docker.com/linux/centos/docker-ce.repo
```

INSTALL DOCKER-CE

- https://mirrors.tuna.tsinghua.edu.cn/

INSTALL DOCKER-CE



INSTALL DOCKER-CE

mirrors.tuna.tsinghua.edu.cn/help/docker-ce/

Open the home page

epel

fedora

flutter

gentoo-portage

gentoo-portage-prefix

git-repo

gitlab-ce

gitlab-ci-multi-runner

gitlab-runner

grafana

hackage

homebrew

Fedora/CentOS/RHEL

以下内容根据 官方文档 修改而来。

如果你之前安装过 docker，请先删掉

```
sudo yum remove docker docker-common docker-selinux docker-engine
```

安装一些依赖

```
sudo yum install -y yum-utils device-mapper-persistent-data lvm2
```

根据你的发行版下载repo文件: CentOS/RHEL

```
wget -O /etc/yum.repos.d/docker-ce.repo https://download.docker.com/linux/centos/docker-ce.repo
```

把软件仓库地址替换为 TUNA:

```
sudo sed -i 's+download.docker.com+mirrors.tuna.tsinghua.edu.cn/docker-ce+' /etc/yum.repos.d/docker-ce.repo
```

最后安装:

```
sudo yum makecache fast
sudo yum install docker-ce
```

INSTALL DOCKER-CE

docs.docker.com/install/linux/linux-postinstall/

docker docs | Search the docs | Guides | Product manuals | Glossary | Reference | Samples

- Get Docker
  - Overview of Docker editions
  - Docker Engine - Community
    - About Docker Engine - Community
  - Cloud
  - Linux
    - CentOS
    - Debian
    - Fedora
    - Ubuntu
    - Binaries
    - **Optional Linux post-installation steps**
  - MacOS
  - Microsoft Windows

Post-installation steps for Linux

Estimated reading time: 16 minutes

This section contains optional procedures for configuring Linux hosts to work better with Docker.

Manage Docker as a non-root user

The Docker daemon binds to a Unix socket instead of a TCP port. By default that Unix socket is owned by the user `root` and other users can only access it using `sudo`. The Docker daemon always runs as the `root` user.

If you don't want to preface the
users to it. When the Docker dae
group.

> **Warning**
>
> The `docker` group grants p
> security in your system, see

To create the `docker` group an

1. Create the `docker` group

Configure Docker to start on boot

Most current Linux distributions (RHEL, CentOS, Fedora, Ubuntu 16.04 and higher) use `systemd` to manage which services start when the system boots. Ubuntu 14.10 and below use `upstart`.

`systemd`

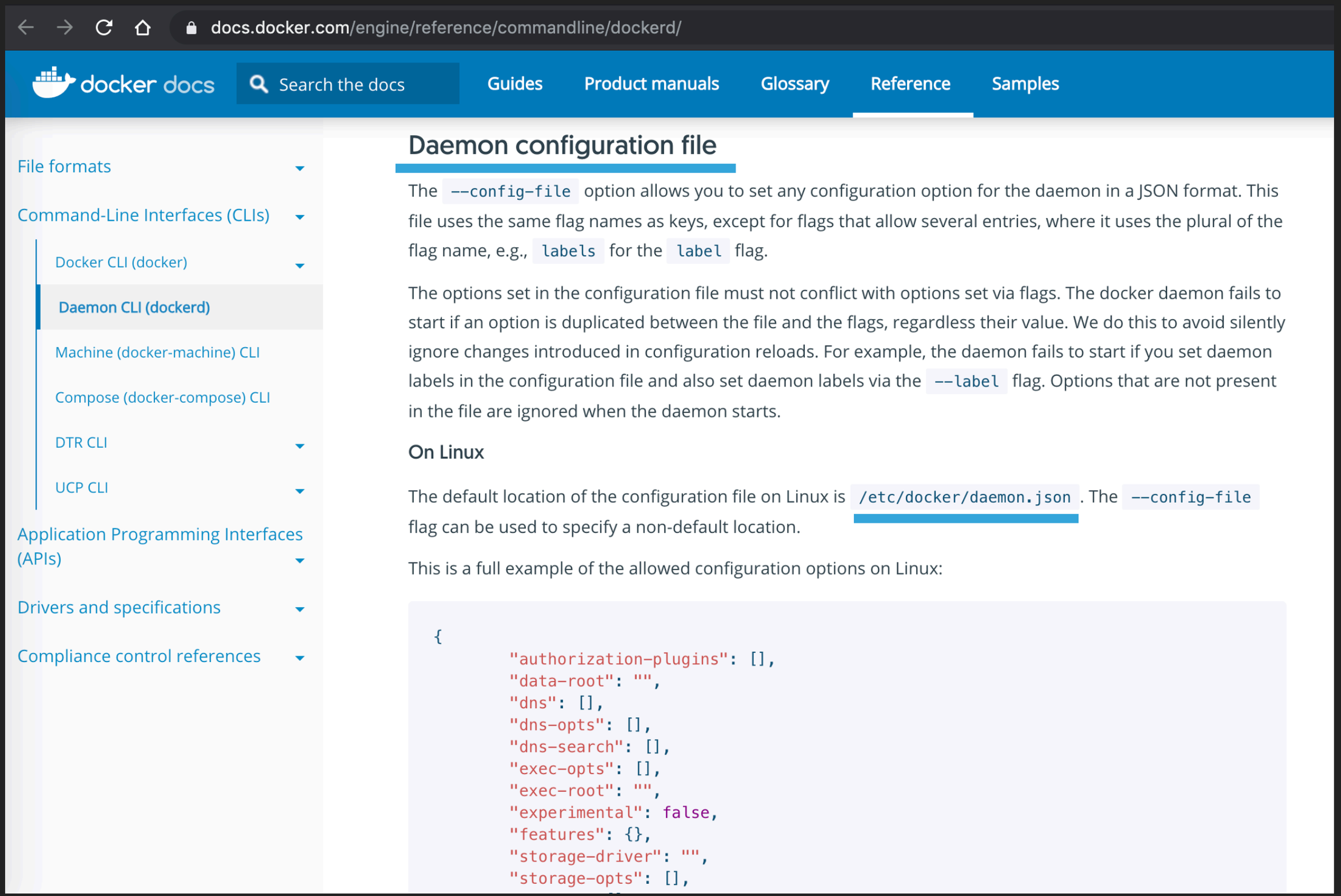
```
$ sudo systemctl enable docker
```

To disable this behavior, use `disable` instead.

```
$ sudo systemctl disable docker
```

If you need to add an HTTP Proxy, set a different directory or partition for the Docker runtime files, or make other customizations, see customize your systemd Docker daemon options.

INSTALL DOCKER-CE



INSTALL DOCKER-CE

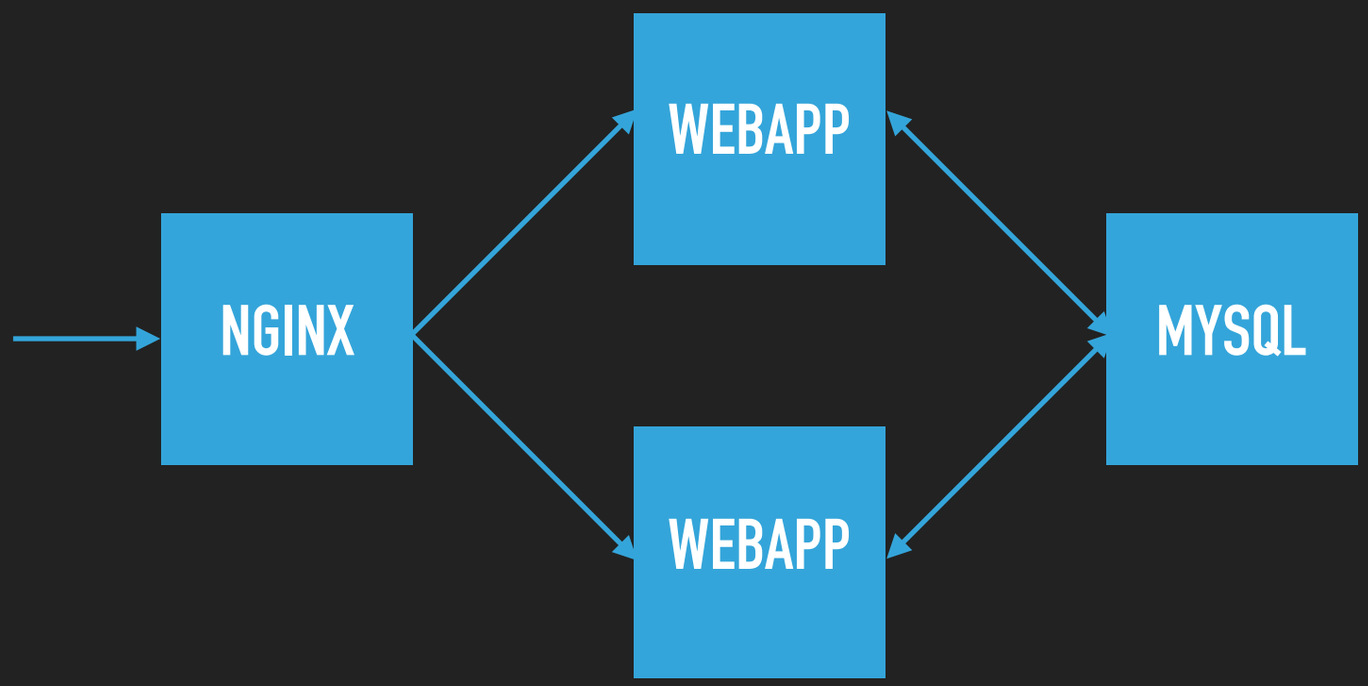
- https://github.com/zenanswer/docker_practice/blob/master/Docker-Install.md

GIT HUB REPO

- https://github.com/zenanswer/docker_practice

DOCKER CLI

- Startup a MySql DB
- Develop one Web APP and start it up
- Startup a Nginx for Load Balance



DOCKER CLI

- MySql DB
  - docker run
    - –name, -p, -v, -e, -d

```
docker run \
--name dp_mysql \
-p 53306:3306 \
-v `pwd`/data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=root \
-e MYSQL_USER=dpuser \
-e MYSQL_PASSWORD=dppassed \
-d mysql:5.6
```

DOCKER CLI

- MySql DB

```
localhost:mysql xcwang$ chmod +x start_up.sh
localhost:mysql xcwang$ ./start_up.sh
Unable to find image 'mysql:5.6' locally


5.6: Pulling from library/mysql
9fc222b64b0a: Downloading [==================>                                ]  8.326MB/22.52MB
291e388076f0: Download complete
d6634415290b: Download complete
1f1e7d852ad4: Download complete
125fc05f36e0: Download complete
02b27e2441e9: Downloading [======================>                            ]  4.685MB/10.17MB
a35058f56a00: Download complete
b43480ce332f: Download complete
5d14b8fc327c: Waiting
423bd0b47bd7: Waiting
189667c449a5: Waiting
```

```
localhost:mysql xcwang$ ./start_up.sh
9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849
localhost:mysql xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                     NAMES
9a5e3439bd8a        mysql:5.6           "docker-entrypoint.s…"   10 seconds ago      Up 9 seconds        0.0.0.0:53306->3306/tcp   dp_mysql
localhost:mysql xcwang$
```

DOCKER CLI

- MySql DB

```
localhost:mysql xcwang$ ls -lt
total 8
drwxr-xr-x@ 8 xcwang  staff  256 Sep  2 13:27 data
-rwxr-xr-x  1 xcwang  staff  174 Sep  2 13:27 start_up.sh
localhost:mysql xcwang$ tree
.
├── data
│   ├── auto.cnf
│   ├── ib_logfile0
│   ├── ib_logfile1
│   ├── ibdata1
│   ├── mysql
│   │   ├── columns_priv.MYD
│   │   ├── columns_priv.MYI
│   │   ├── columns_priv.frm
│   │   ├── db.MYD
│   │   ├── db.MYI
│   │   ├── db.frm
│   │   ├── event.MYD
│   │   ├── event.MYI
│   │   ├── event.frm
│   │   ├── func.MYD
│   │   ├── func.MYI
│   │   ├── func.frm
│   │   ├── general_log.CSM
│   │   ├── general_log.CSV
│   │   ├── general_log.frm
```

DOCKER CLI

- MySql DB
  - docker images

```
localhost:mysql xcwang$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mysql               5.6                 732765f8c7d2        2 weeks ago         257MB
ubuntu              18.04               3556258649b2        5 weeks ago         64.2MB
golang              1.11                ee3ec9ac0398        6 weeks ago         796MB
debian              10.0                00bf7fdd8baf        7 weeks ago         114MB
ubuntu              latest              4c108a37151f        2 months ago        64.2MB
node                10.15.3-alpine      56bc3a1ed035        3 months ago        71MB
jhipster/jhipster   latest              34a661b71fd0        3 months ago        1.41GB
cassandra           3.11.4              a05e8a072b59        3 months ago        323MB
jboss/wildfly       10.1.0.Final        d1c81a71a5ee        6 months ago        676MB
localhost:mysql xcwang$
```

DOCKER CLI

- MySql DB
  - docker inspect

```
localhost:mysql xcwang$ docker inspect dp_mysql
[
    {
        "Id": "9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849",
        "Created": "2019-09-02T05:27:18.3656679Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 3977,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2019-09-02T05:27:19.110067Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:732765f8c7d21d1a7ba832e444df1116959b3ba6d134abe20cda5b5e0013
        "ResolvConfPath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795c
        "HostnamePath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf9
        "HostsPath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf9820
        "LogPath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf982076
        "Name": "/dp_mysql",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
```

DOCKER CLI

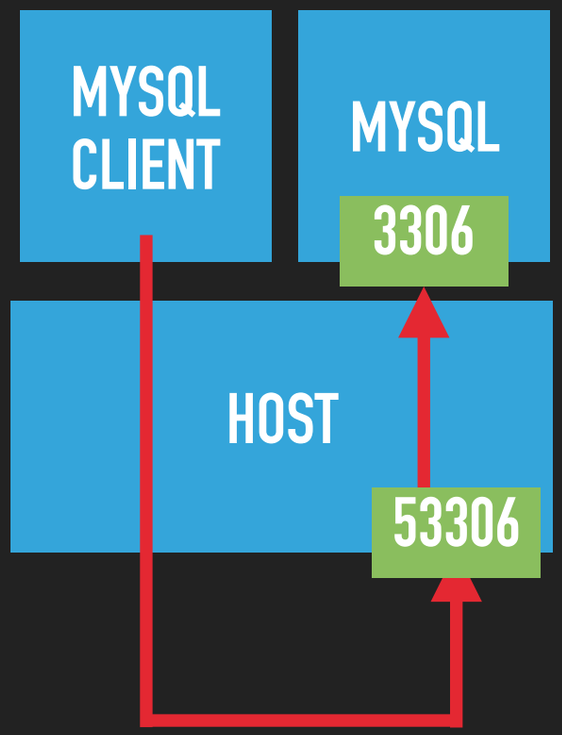
- MySql DB
  - docker ps and docker exec

```
localhost:mysql xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
9a5e3439bd8a        mysql:5.6           "docker-entrypoint.s…"   7 minutes ago       Up 7 minutes        0.0.0.0:53306->3306/tcp  dp_mysql
localhost:mysql xcwang$ docker exec -it 9a bash
root@9a5e3439bd8a:/# ll
bash: ll: command not found
root@9a5e3439bd8a:/# ls -lt
total 68
drwx------   1 root root 4096 Sep  2 05:34 root
drwxrwxrwt   1 root root 4096 Sep  2 05:27 tmp
drwxr-xr-x   5 root root  340 Sep  2 05:27 dev
dr-xr-xr-x  13 root root    0 Sep  2 05:27 sys
dr-xr-xr-x 185 root root    0 Sep  2 05:27 proc
drwxr-xr-x   1 root root 4096 Sep  2 05:27 etc
lrwxrwxrwx   1 root root   34 Aug 14 06:10 entrypoint.sh -> usr/local/bin/docker-entrypoint.sh
drwxr-xr-x   1 root root 4096 Aug 14 06:09 run
drwxr-xr-x   1 root root 4096 Aug 14 06:09 usr
drwxr-xr-x   1 root root 4096 Aug 14 06:09 bin
drwxr-xr-x   2 root ro
drwxr-xr-x   1 root ro
drwxr-xr-x   2 root ro
drwxr-xr-x   2 root ro
```

```
localhost:mysql xcwang$ docker exec dp_mysql echo 'Hello world!'
Hello world!
localhost:mysql xcwang$
```

DOCKER CLI

- MySql Client




```
localhost:mysql xcwang$ cat mysql_cli.sh
docker run -it --rm mysql:5.6 mysql -h192.168.31.221 -P53306 -udpuser -pdppassed
localhost:mysql xcwang$
```

```
localhost:mysql xcwang$ ./mysql_cli.sh
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.45 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
localhost:mysql xcwang$ $
```

DOCKER CLI

- MySql DB
  - docker stop and rm

```
localhost:mysql xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                     NAMES
9a5e3439bd8a        mysql:5.6           "docker-entrypoint.s…"   46 minutes ago      Up 46 minutes       0.0.0.0:53306->3306/tcp   dp_mysql
localhost:mysql xcwang$ docker stop 9a5e3439bd8a
9a5e3439bd8a
localhost:mysql xcwang$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
9a5e3439bd8a        mysql:5.6           "docker-entrypoint.s…"   46 minutes ago      Exited (0) 11 seconds ago                       dp_mysql
localhost:mysql xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
localhost:mysql xcwang$ docker rm 9a5e3439bd8a
9a5e3439bd8a
localhost:mysql xcwang$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
localhost:mysql xcwang$
```

https://docs.docker.com/engine/reference/run/

DOCKERFILE

- WebAPP
  - Dockerfile

```
localhost:webapp xcwang$ tree
.
├── Dockerfile
├── build.sh
├── node-mysql-crud-app
│   ├── README.md
│   ├── app.js
│   ├── package-lock.json
│   ├── package.json
│   ├── routes
│   │   ├── index.js
│   │   └── player.js
│   └── views
│       ├── add-player.ejs
│       ├── edit-player.ejs
│       ├── index.ejs
│       └── partials
│           └── header.ejs
├── public
│   └── assets
│       └── img
└── run.sh

7 directories, 13 files
```

Dockerfile

```
1   FROM node:10.15.3-alpine
2
3   ADD node-mysql-crud-app /node-mysql-crud-app
4
5   WORKDIR /node-mysql-crud-app
6
7   RUN mkdir -p public/assets/img/ && \
8       npm install
9
10  ENV DB_URL='mysql://user:passwd@localhost:port/database' \
11      WEB_PORT=80
12
13  ENTRYPOINT [ "node", "app.js" ]
14
15  EXPOSE 80/tcp
16
```

docker build ./ -f Dockerfile -t dp_webapp

https://docs.docker.com/engine/reference/builder/

DOCKERFILE

- WebAPP
  - Dockerfile & docker build

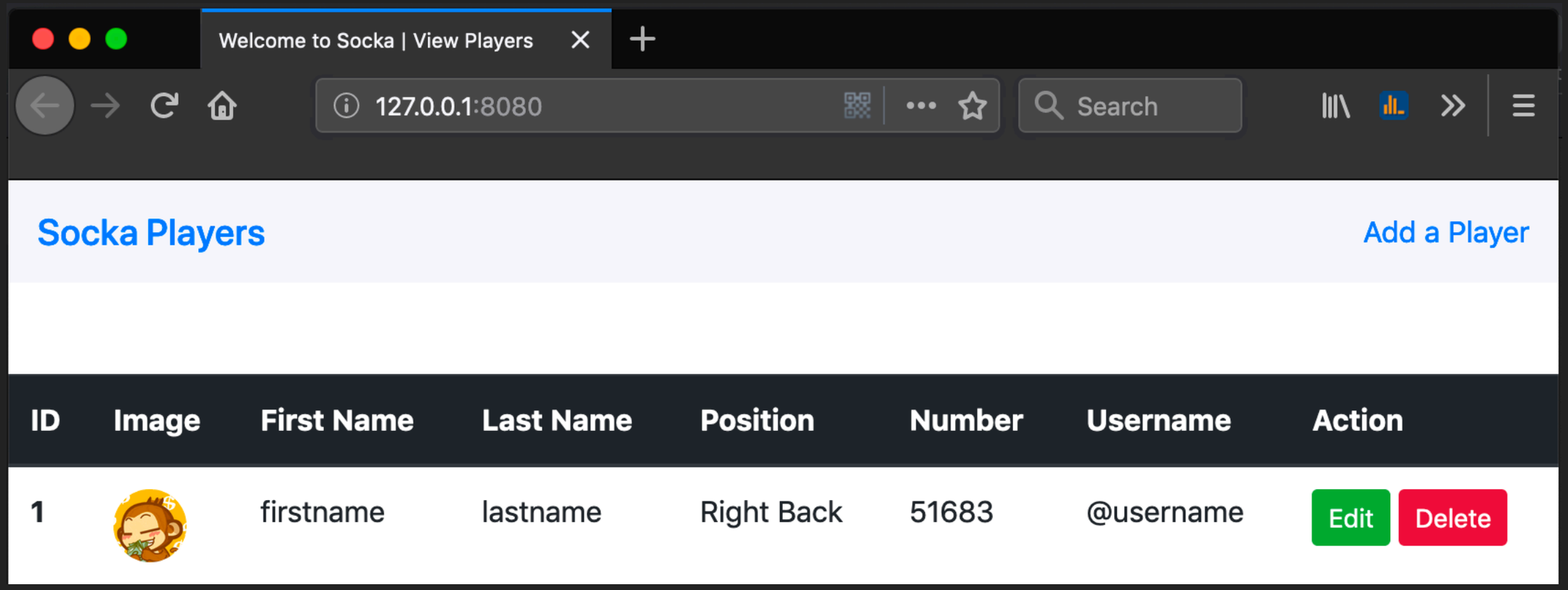
```
docker build ./ -f Dockerfile -t dp_webapp
```

```
localhost:webapp xcwang$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dp_webapp           latest              9a1989ad8ca2        34 minutes ago      79.3MB
<none>              <none>              245e28adc24a        41 minutes ago      79.3MB
nginx               1.17.3              5a3221f0137b        2 weeks ago         126MB
mysql               5.6                 732765f8c7d2        2 weeks ago         257MB
ubuntu              18.04               3556258649b2        5 weeks ago         64.2MB
golang              1.11                ee3ec9ac0398        6 weeks ago         796MB
debian              10.0                00bf7fdd8baf        7 weeks ago         114MB
ubuntu              latest              4c108a37151f        2 months ago        64.2MB
node                10.15.3-alpine      56bc3a1ed035        3 months ago        71MB
jhipster/jhipster   latest              34a661b71fd0        3 months ago        1.41GB
cassandra           3.11.4              a05e8a072b59        3 months ago        323MB
jboss/wildfly       10.1.0.Final        d1c81a71a5ee        6 months ago        676MB
localhost:webapp xcwang$
```

DOCKERFILE

- WebAPP
  - docker run

docker run -p 8080:80 -e DB_URL=mysql://dpuser:dppassed@192.168.31.221:53306/dp dp_webapp



DOCKERFILE

- WebAPP
  - https://dev.to/achowba/build-a-simple-app-using-node-js-and-mysql-19me
  - https://github.com/achowba/node-mysql-crud-app

DOCKERCLI

- NGINX

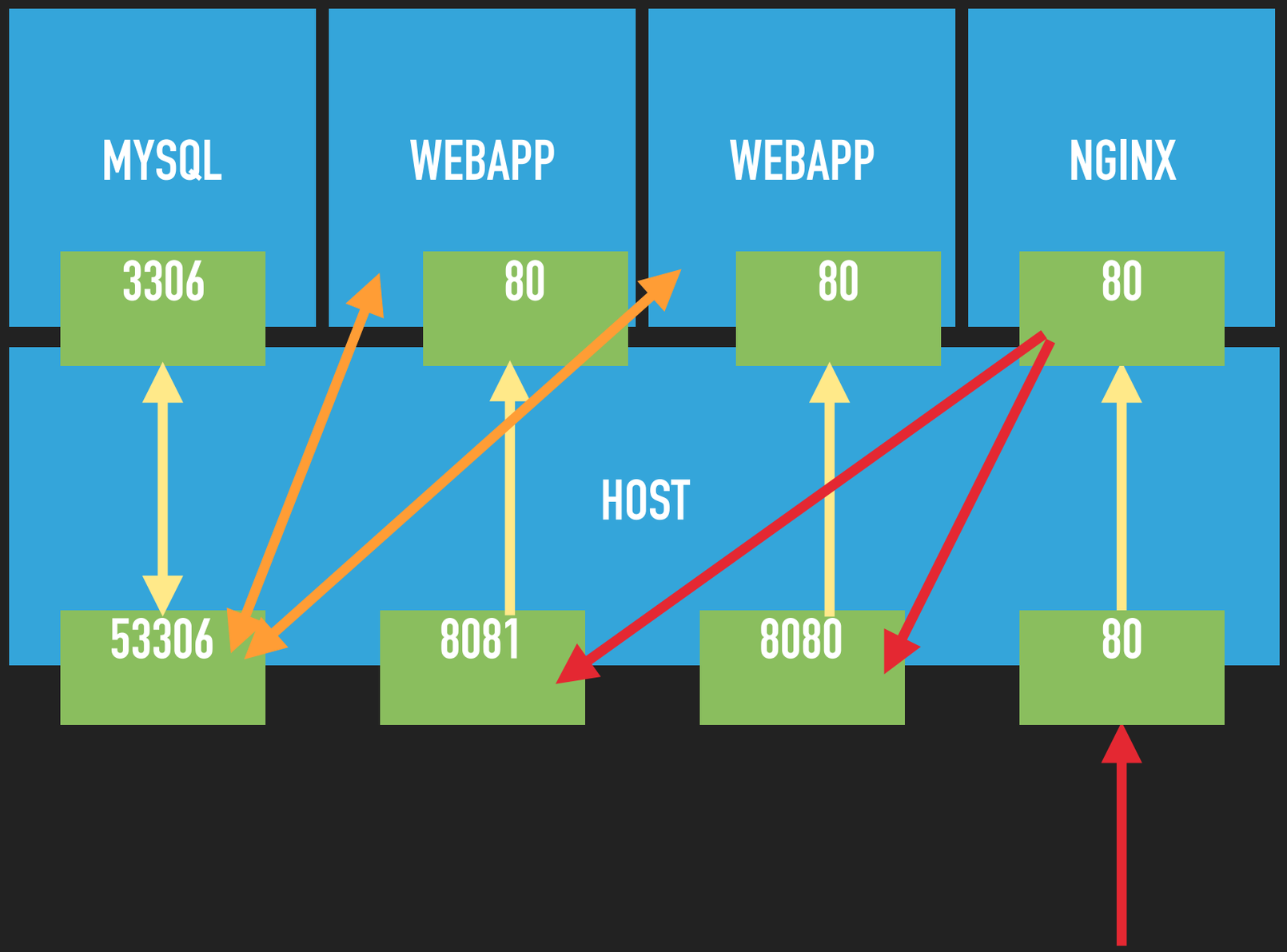
```
localhost:nginx xcwang$ cat nginx.conf
http {
        upstream backend {
                least_conn;
                server 192.168.31.221:8080;
                server 192.168.31.221:8081;
        }

        server {
                listen 80;
                location / {
                        proxy_pass http://backend;
                }
        }
}

events {
  worker_connections  1024;  ## Default: 1024
}
localhost:nginx xcwang$
```

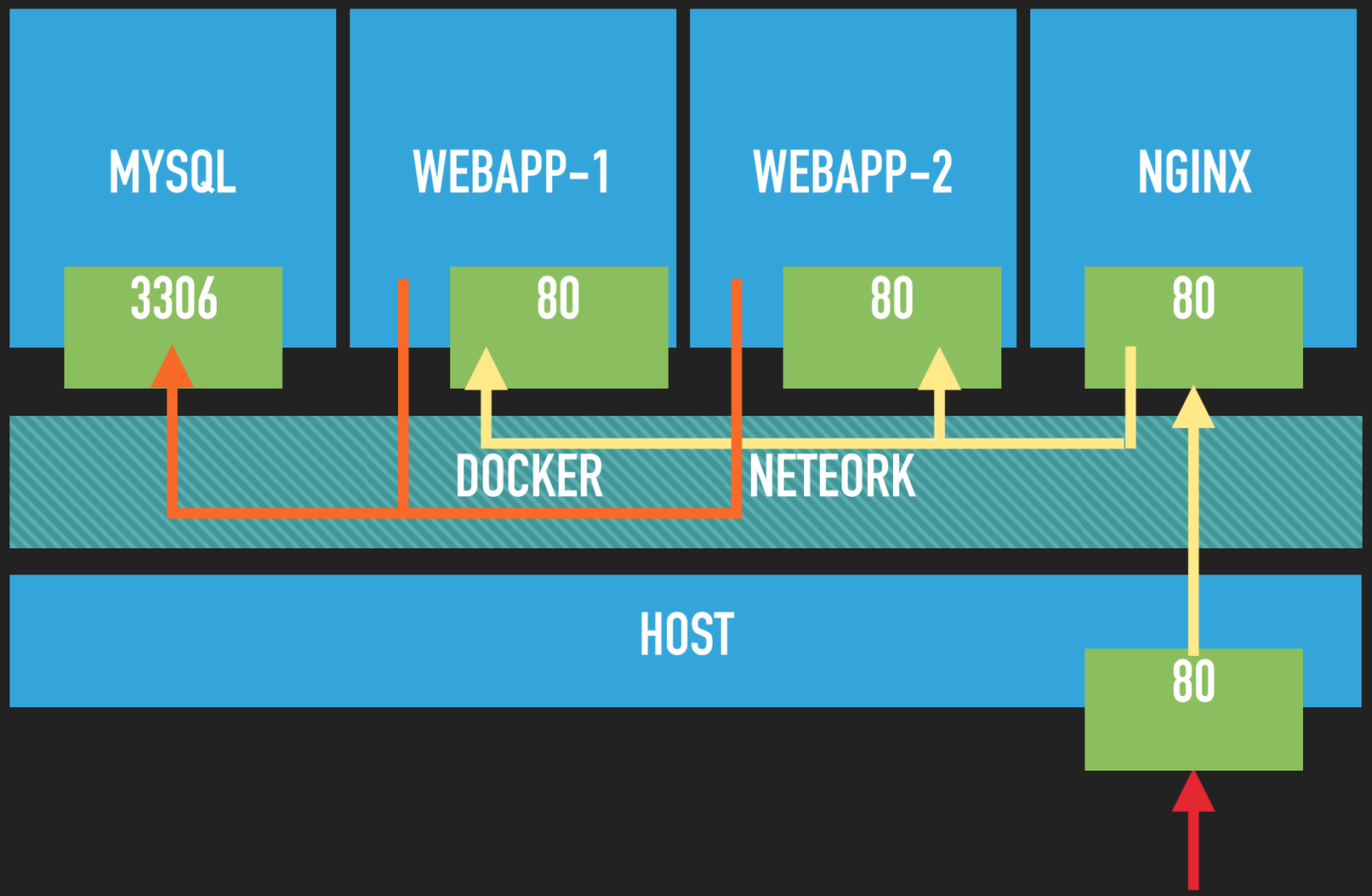
```
docker run \
-v `pwd`/nginx.conf:/etc/nginx/nginx.conf \
-p 80:80 \
nginx:1.17.3
```

DOCKERCLI


MYSQL
WEBAPP
WEBAPP
NGINX
3306
80
80
80
HOST
53306
8081
8080
80

DOCKER COMPOSE & CLI

- MySql DB + 2 * Web APP + Nginx



DOCKER - HOW TO

DOCKER COMPOSE & CLI

- Docker compose file

docker-compose.yaml

```
version: "3.2"
services:
  mysql:
    image: mysql:5.6
    restart: on-failure
    volumes:
      - type: bind
        source: ./data/mysql
        target: /var/lib/mysql
      - type: bind
        source: ../webapp/node-mysql-crud-app/create_table.sql
        target: /docker-entrypoint-initdb.d/create_table.sql
    environment:
      - MYSQL_ROOT_PASSWORD=root
      - MYSQL_USER=dpuser
      - MYSQL_PASSWORD=dppassed
      - MYSQL_DATABASE=dp
  webapp:
    image: dp_webapp
    restart: on-failure
    volumes:
      - type: bind
        source: ./data/webapp
        target: /node-mysql-crud-app/public/assets/img
    environment:
      - DB_URL=mysql://dpuser:dppassed@mysql/dp
  nginx:
    image: nginx:1.17.3
    ports:
    - "80:80"
    restart: on-failure
    volumes:
      - type: bind
        source: ./nginx.conf
        target: /etc/nginx/nginx.conf
```

nginx.conf

```
http {
        upstream backend {
                least_conn;
                server webapp;
        }

        server {
                listen 80;
                location / {
                        proxy_pass http://backend;
                }
        }
}

events {
  worker_connections  1024;  ## Default: 1024
}
```

DOCKER COMPOSE & CLI

- compose up & down

```
localhost:docker xcwang$ docker-compose up -d
Creating network "docker_default" with the default driver
Creating docker_nginx_1  ... done
Creating docker_webapp_1 ... done
Creating docker_mysql_1  ... done
localhost:docker xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                          PORTS                   NAMES
2fb11984e8be        mysql:5.6           "docker-entrypoint.s…"   4 seconds ago       Up 2 seconds                    3306/tcp                docker_mysql_1
048c8be2410d        nginx:1.17.3        "nginx -g 'daemon of…"   4 seconds ago       Up 2 seconds                    0.0.0.0:80->80/tcp      docker_nginx_1
6d03049c93ce        dp_webapp           "node app.js"            4 seconds ago       Restarting (1) 2 seconds ago                            docker_webapp_1
localhost:docker xcwang$ docker-compose ps
     Name                   Command             State         Ports
-------------------------------------------------------------------------
docker_mysql_1    docker-entrypoint.sh mysqld   Up      3306/tcp
docker_nginx_1    nginx -g daemon off;          Up      0.0.0.0:80->80/tcp
docker_webapp_1   node app.js                   Up      80/tcp
```

```
localhost:docker xcwang$ docker-compose down
Stopping docker_mysql_1  ... done
Stopping docker_nginx_1  ... done
Stopping docker_webapp_1 ... done
Removing docker_mysql_1  ... done
Removing docker_nginx_1  ... done
Removing docker_webapp_1 ... done
Removing network docker_default
localhost:docker xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
localhost:docker xcwang$
```

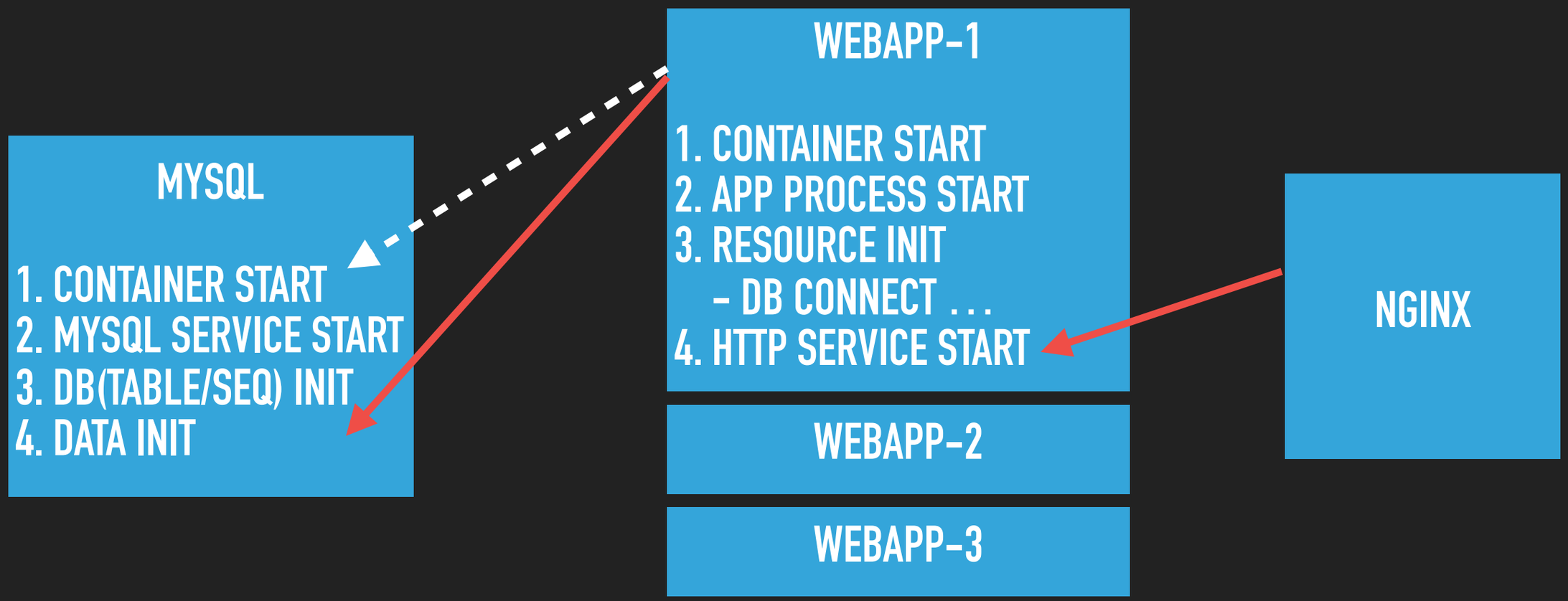
DOCKER COMPOSE & CLI

- compose logs

```
localhost:docker xcwang$ docker-compose logs -f webapp
Attaching to docker_webapp_1
webapp_1  | Server running on port: 80
webapp_1  | /node-mysql-crud-app/app.js:39
webapp_1  |         throw err;
webapp_1  |         ^
webapp_1  |
webapp_1  | Error: connect ECONNREFUSED 172.24.0.4:3306
webapp_1  |     at TCPConnectWrap.afterConnect [as oncomplete] (net.js:1097:14)
webapp_1  |     --------------------
webapp_1  |     at Protocol._enqueue (/node-mysql-crud-app/node_modules/mysql/lib/protocol/Protocol.js:145:48)
webapp_1  |     at Protocol.handshake (/node-mysql-crud-app/node_modules/mysql/lib/protocol/Protocol.js:52:23)
webapp_1  |     at Connection.connect (/node-mysql-crud-app/node_modules/mysql/lib/Connection.js:130:18)
webapp_1  |     at Object.<anonymous> (/node-mysql-crud-app/app.js:37:4)
webapp_1  |     at Module._compile (internal/modules/cjs/loader.js:701:30)
webapp_1  |     at Object.Module._extensions..js (internal/modules/cjs/loader.js:712:10)
webapp_1  |     at Module.load (internal/modules/cjs/loader.js:600:32)
webapp_1  |     at tryModuleLoad (internal/modules/cjs/loader.js:539:12)
webapp_1  |     at Function.Module._load (internal/modules/cjs/loader.js:531:3)
webapp_1  |     at Function.Module.runMain (internal/modules/cjs/loader.js:754:12)
webapp_1  | Server running on port: 80
webapp_1  | Connected to database
webapp_1  | (node:1) [DEP0096] DeprecationWarning: timers.unenroll() is deprecated. Please use clearTimeout instead.
```

DOCKER COMPOSE & CLI

- healthcheck & depends_on



DOCKER COMPOSE & CLI

- healthcheck & depends_on

```yaml
webapp:
  image: dp_webapp
  restart: on-failure
  volumes:
    - type: bind
      source: ./data/webapp
      target: /node-mysql-crud-app/public/assets/img
    - type: bind
      source: ./healthcheck.js
      target: /healthcheck.js
  environment:
    - DB_URL=mysql://dpuser:dppassed@mysql/dp
  healthcheck:
    # test: ["CMD", "curl", "http://localhost/"]
    test: ["CMD", "node", "/healthcheck.js"]
    start_period: 10s
    interval: 2s
    timeout: 3s
    retries: 4
  depends_on:
    mysql:
      condition: service_healthy # or service_started
```

```yaml
mysql:
  image: mysql:5.6
  restart: on-failure
  volumes:
    - type: bind
      source: ./data/mysql      You, 17 hours ago • docker-compose w
      target: /var/lib/mysql
    - type: bind
      source: ../webapp/node-mysql-crud-app/create_table.sql
      target: /docker-entrypoint-initdb.d/create_table.sql
    - type: bind
      source: ./checkiftablexists.sh
      target: /checkiftablexists.sh
  environment:
    - MYSQL_ROOT_PASSWORD=root
    - MYSQL_USER=dpuser
    - MYSQL_PASSWORD=dppassed
    - MYSQL_DATABASE=dp
  healthcheck:
    test: ["CMD", "bash", "/checkiftablexists.sh" ,"dp", "players"]
    start_period: 30s
    interval: 5s
    timeout: 3s
    retries: 4
```

DOCKER COMPOSE & CLI

- healthcheck & depends_on

```
localhost:~ xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                             PORTS               NAMES
8d5843e52d8b        mysql:5.6           "docker-entrypoint.s…"   5 seconds ago       Up 3 seconds (health: starting)    3306/tcp            docker_mysql_1
```

```
localhost:~ xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                     PORTS               NAMES
8d5843e52d8b        mysql:5.6           "docker-entrypoint.s…"   8 seconds ago       Up 6 seconds (healthy)     3306/tcp            docker_mysql_1
```

```
localhost:~ xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                             PORTS               NAMES
ec142d74e44c        dp_webapp           "node app.js"            3 seconds ago       Up 2 seconds (health: starting)    80/tcp              docker_webapp_1
8d5843e52d8b        mysql:5.6           "docker-entrypoint.s…"   10 seconds ago      Up 8 seconds (healthy)             3306/tcp            docker_mysql_1
```

```
localhost:~ xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                     PORTS                  NAMES
1691fd6b170d        nginx:1.17.3        "nginx -g 'daemon of…"   2 seconds ago       Up Less than a second      0.0.0.0:80->80/tcp     docker_nginx_1
ec142d74e44c        dp_webapp           "node app.js"            5 seconds ago       Up 4 seconds (healthy)     80/tcp                 docker_webapp_1
8d5843e52d8b        mysql:5.6           "docker-entrypoint.s…"   12 seconds ago      Up 10 seconds (healthy)    3306/tcp               docker_mysql_1
```

```
localhost:~ xcwang$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                     PORTS                  NAMES
1691fd6b170d        nginx:1.17.3        "nginx -g 'daemon of…"   7 seconds ago       Up 6 seconds               0.0.0.0:80->80/tcp     docker_nginx_1
ec142d74e44c        dp_webapp           "node app.js"            10 seconds ago      Up 9 seconds (healthy)     80/tcp                 docker_webapp_1
8d5843e52d8b        mysql:5.6           "docker-entrypoint.s…"   17 seconds ago      Up 15 seconds (healthy)    3306/tcp               docker_mysql_1
```

DOCKER COMPOSE & CLI

- compose V2 v.s. V3
  - docker-compose
    - https://docs.docker.com/compose/compose-file/compose-file-v2/
- docker swarm/stack
  - https://docs.docker.com/compose/compose-file/

DOCKER SWARM