

SAY HI TO VIRTUALIZATION

---

# DOCKER

Wang Xuechen

2019/09/02

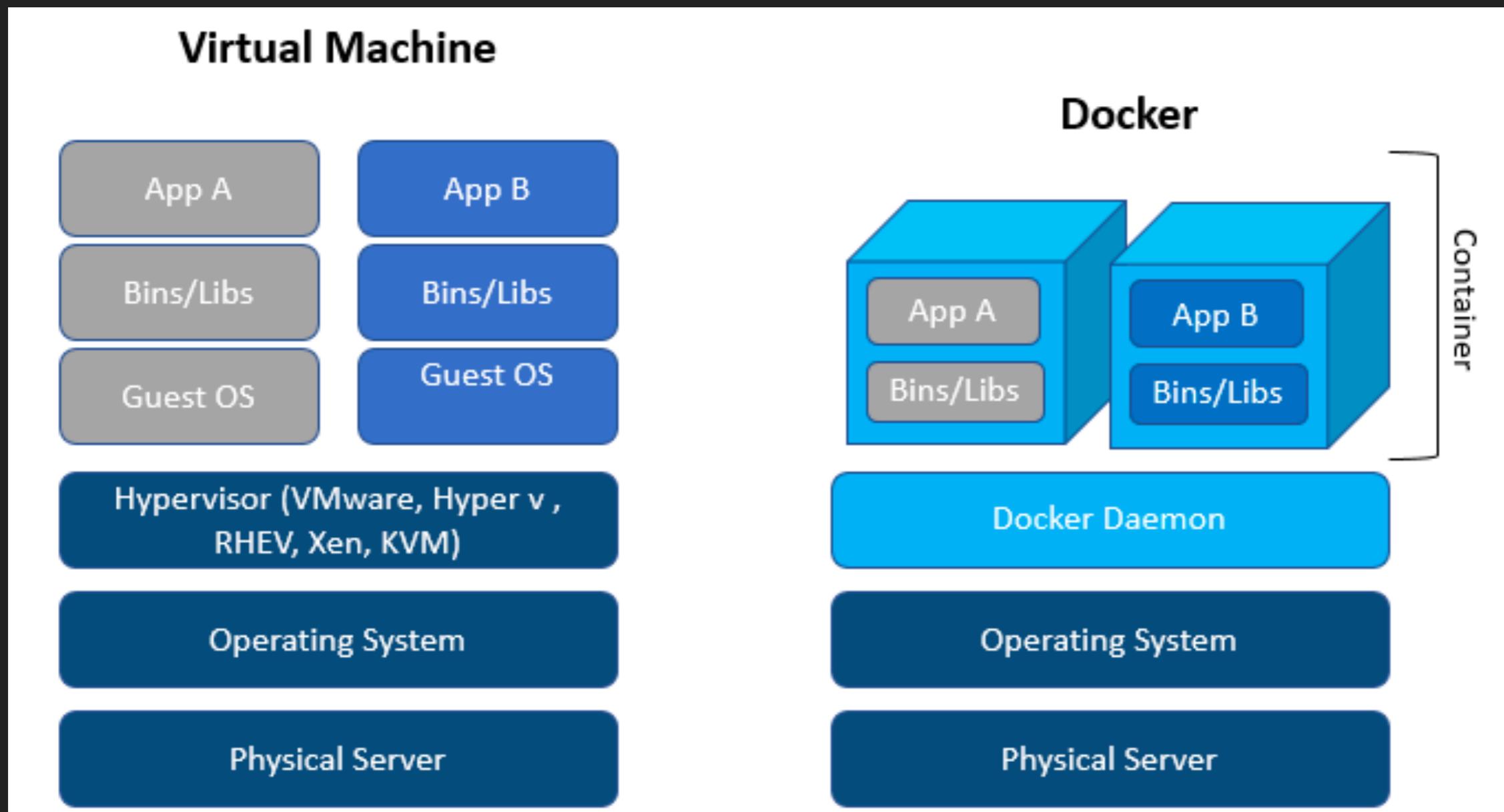
## AGENDA

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

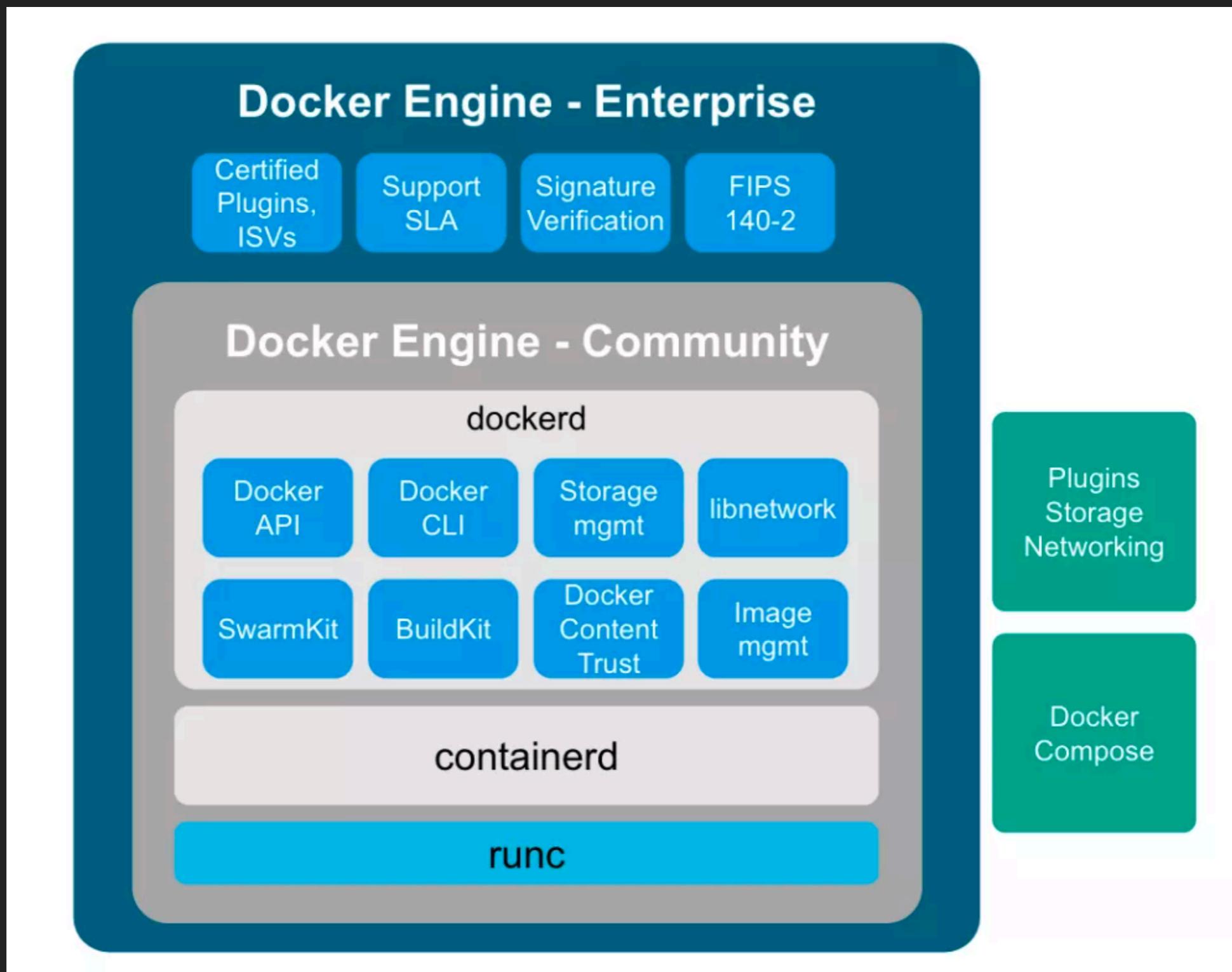
# AGENDA

- ▶ How to
  - ▶ Docker CLI (`docker`) / Dockerfile
  - ▶ Docker Compose File & CLI
  - ▶ Docker Swarm\*

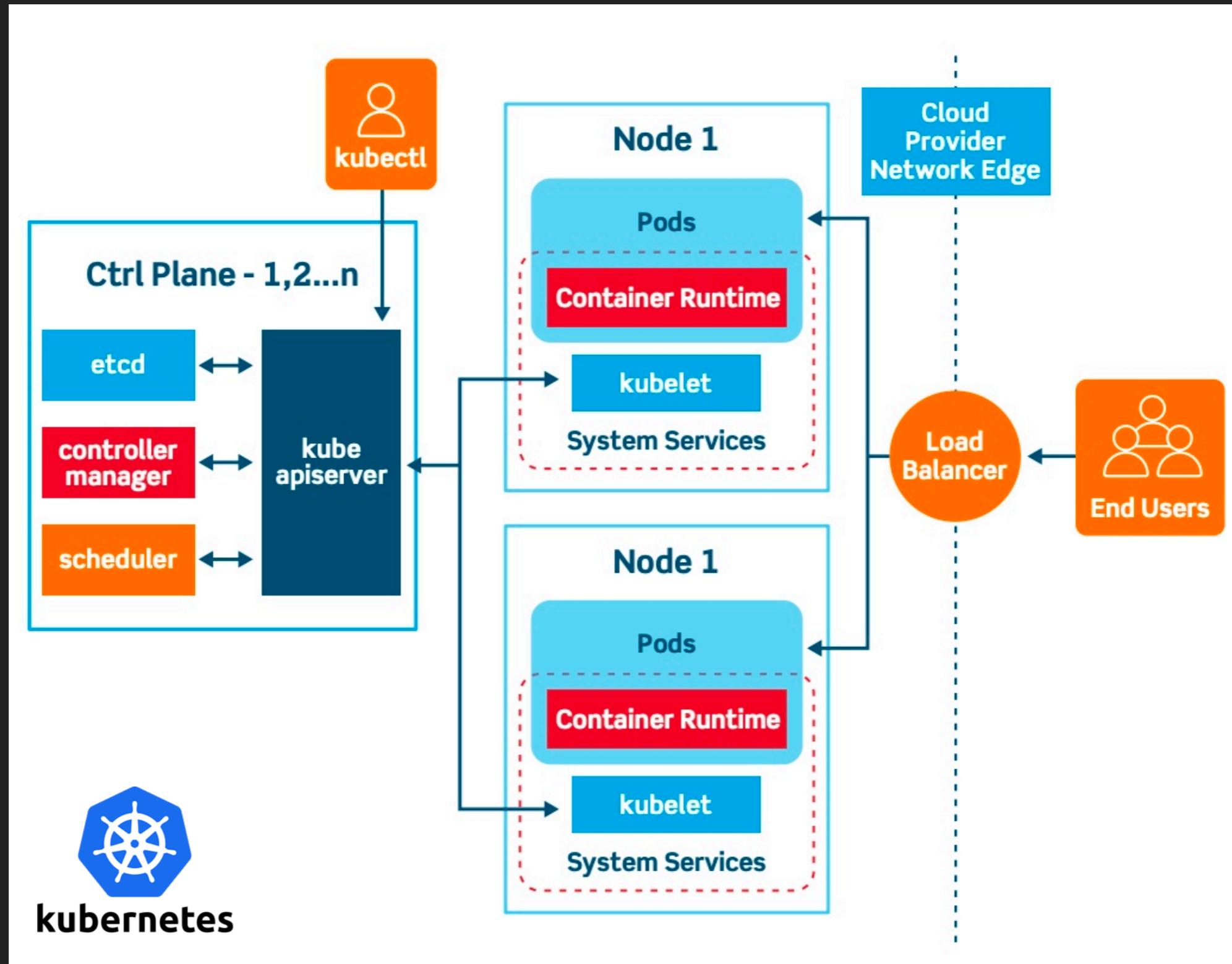
## DIFFERENT



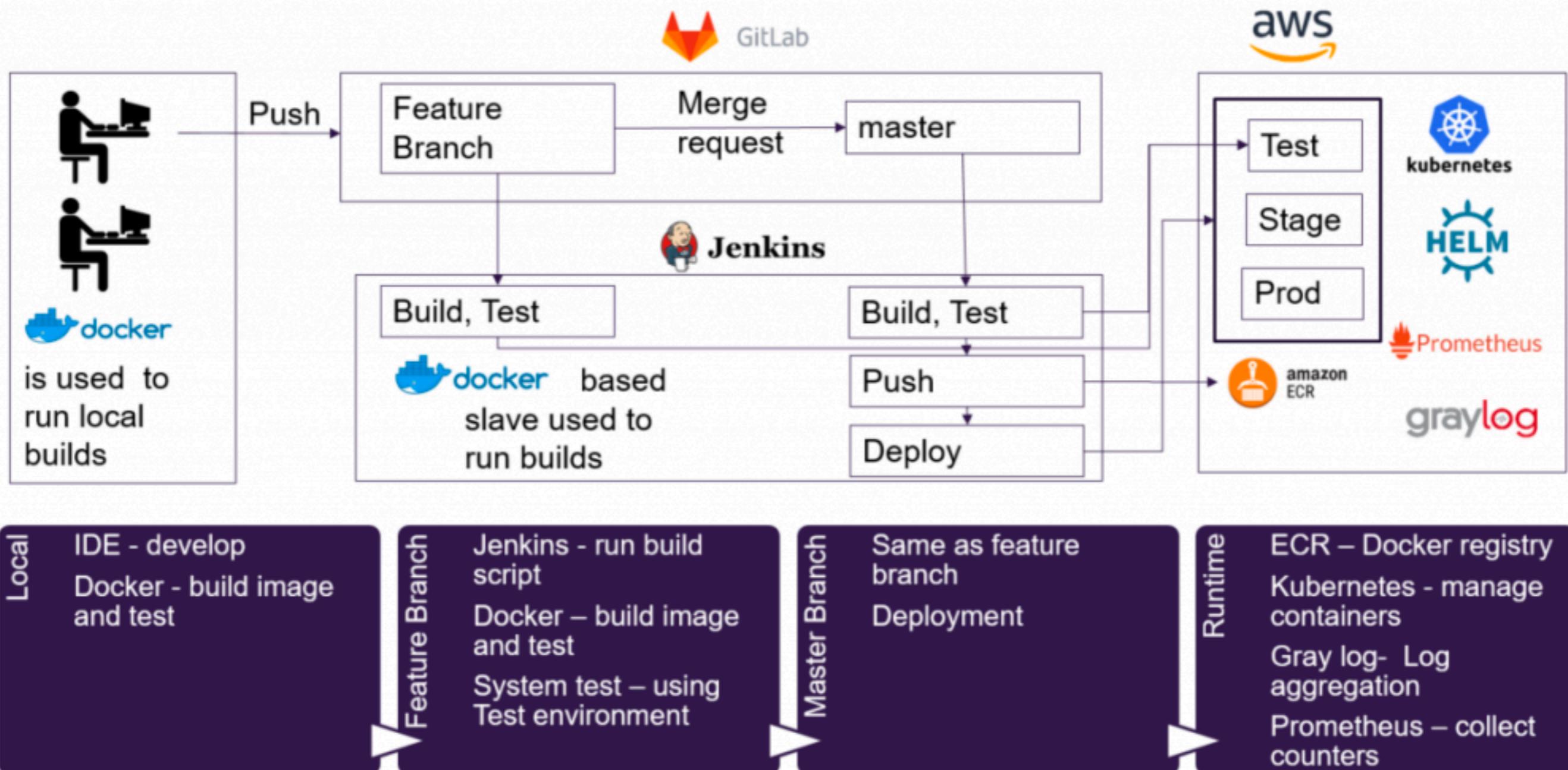
# ARCHITECTURE



## K8S

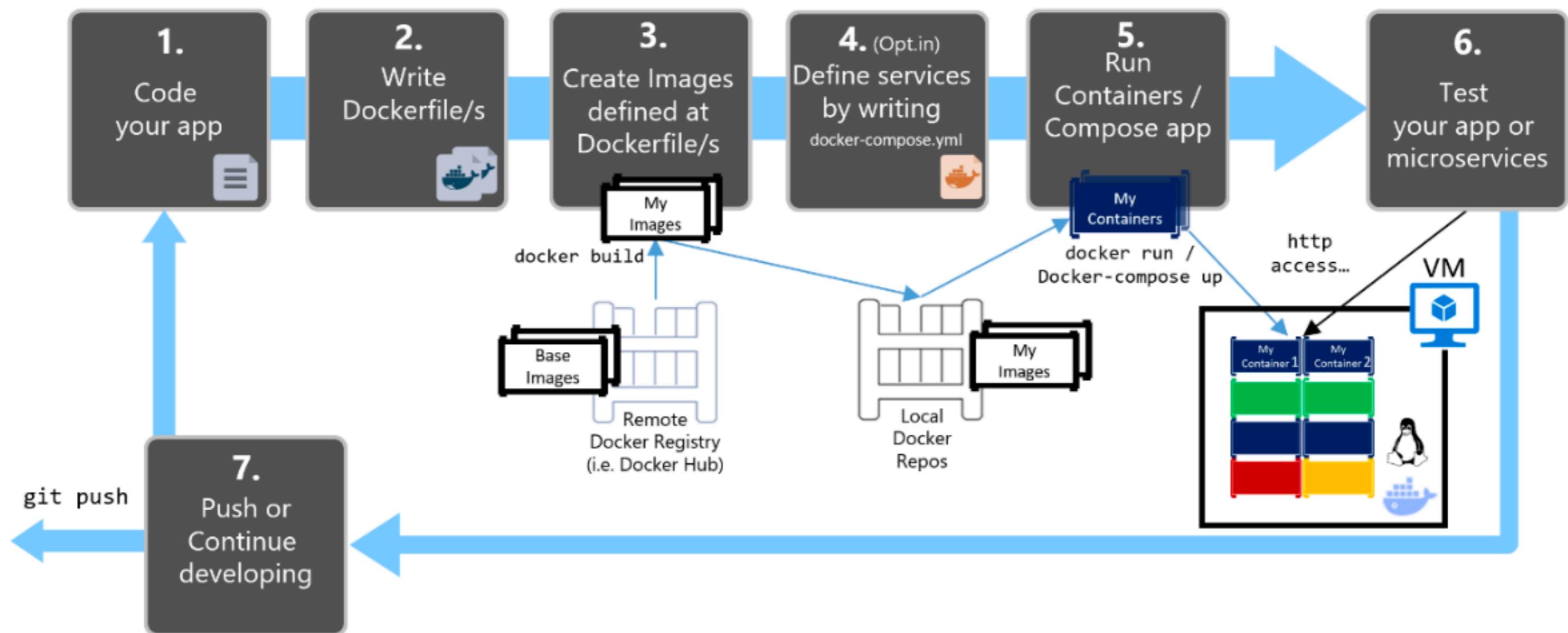


## WORKFLOW

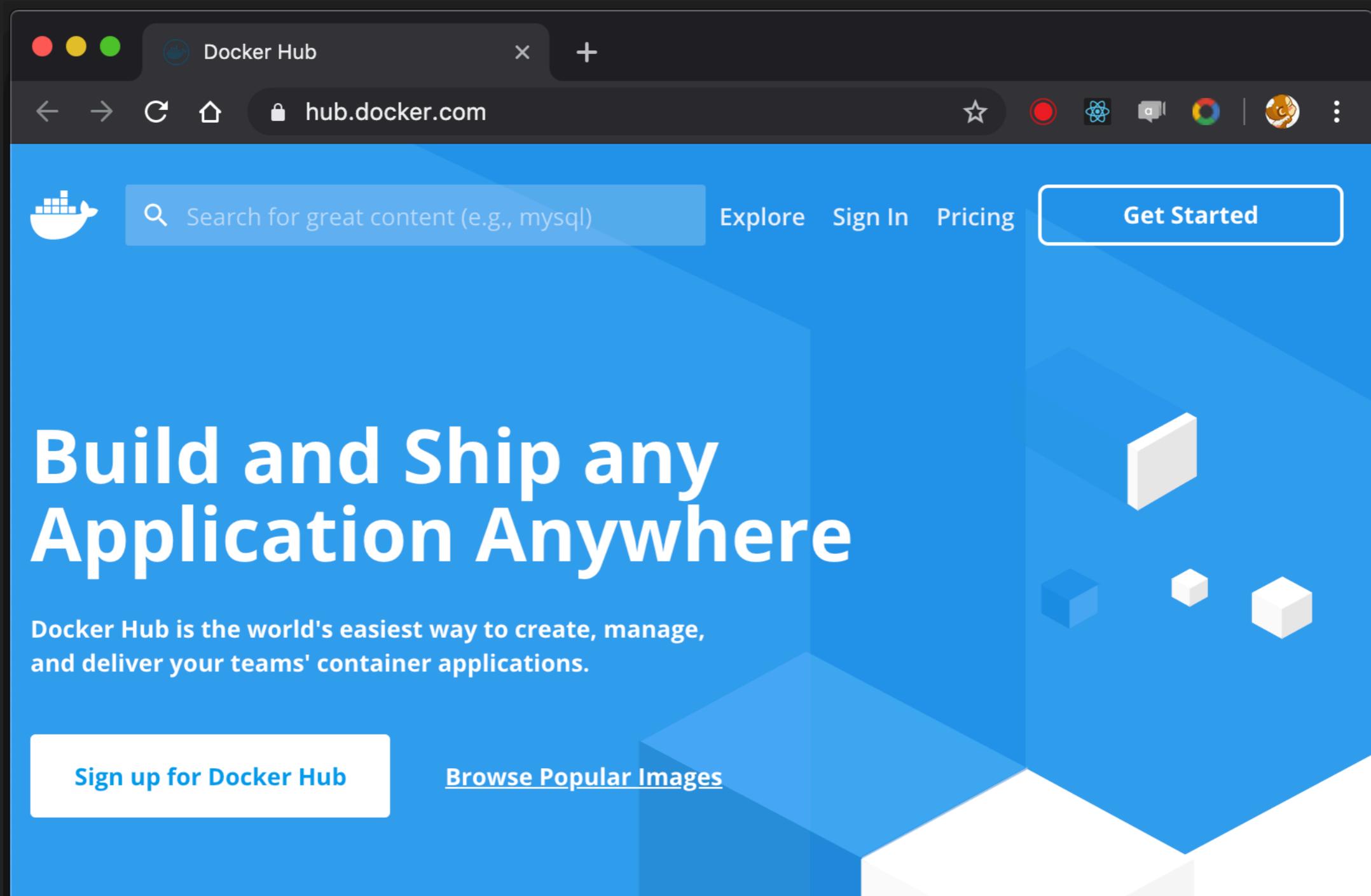


## WORKFLOW

Inner-Loop development workflow for Docker apps



# DOCKER HUB



# DOCKER - WHAT'S DOCKER

## DOCKER HUB

Docker Certified [i](#)

Docker Certified

Images

Verified Publisher [i](#)  
*Docker Certified And  
Verified Publisher  
Content*

Official Images [i](#)  
*Official Images  
Published By Docker*

Categories [i](#)

Analytics

Application Frameworks

Application Infrastructure

Application Services

Base Images

OFFICIAL IMAGE 

mysql   
Updated an hour ago

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

MariaDB is a community-developed fork of MySQL intended to rem...

10M+ 3.0K  
Downloads Stars

# DOCKER - WHAT'S DOCKER

## DOCKER HUB

**OFFICIAL IMAGE** 

 **centos**  
Updated 2 minutes ago

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

PostgreSQL is an advanced Object-Relational database manageme...

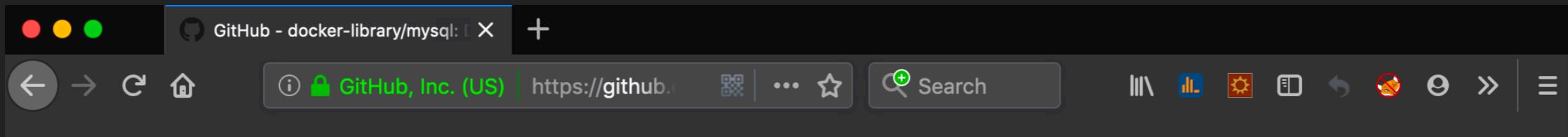
Container Linux x86-64

 **centos/mariadb-101-centos7**  
By [centos](#) • Updated 5 months ago

MariaDB 10.1 SQL database server

Container Linux x86-64

# DOCKER HUB & GIT HUB



The screenshot shows a web browser window with the GitHub URL <https://github.com/docker-library/mysql> in the address bar. The page content is as follows:

**<https://github.com/docker-library/mysql>**

**Maintained by: the Docker Community and the MySQL Team**

This is the Git repo of the Docker "Official Image" for `mysql` (not to be confused with any official `mysql` image provided by `mysql` upstream). See [the Docker Hub page](#) for the full readme on how to use this Docker image and for information regarding contributing and issues.

The [full image description on Docker Hub](#) is generated/maintained over in [the docker-library/docs repository](#), specifically in [the mysql directory](#).

**See a change merged here that doesn't show up on Docker Hub yet?**

For more information about the full official images change lifecycle, see [the "An image's source changed in Git, now what?" FAQ entry](#).

For outstanding `mysql` image PRs, check [PRs with the "library/mysql" label on the official-images repository](#). For the current "source of truth" for `mysql`, see [the library/mysql file in the official-images repository](#).

---

- [Travis CI passing](#)
- [Automated update.sh passing](#)

## DOCKER HUB & GIT HUB

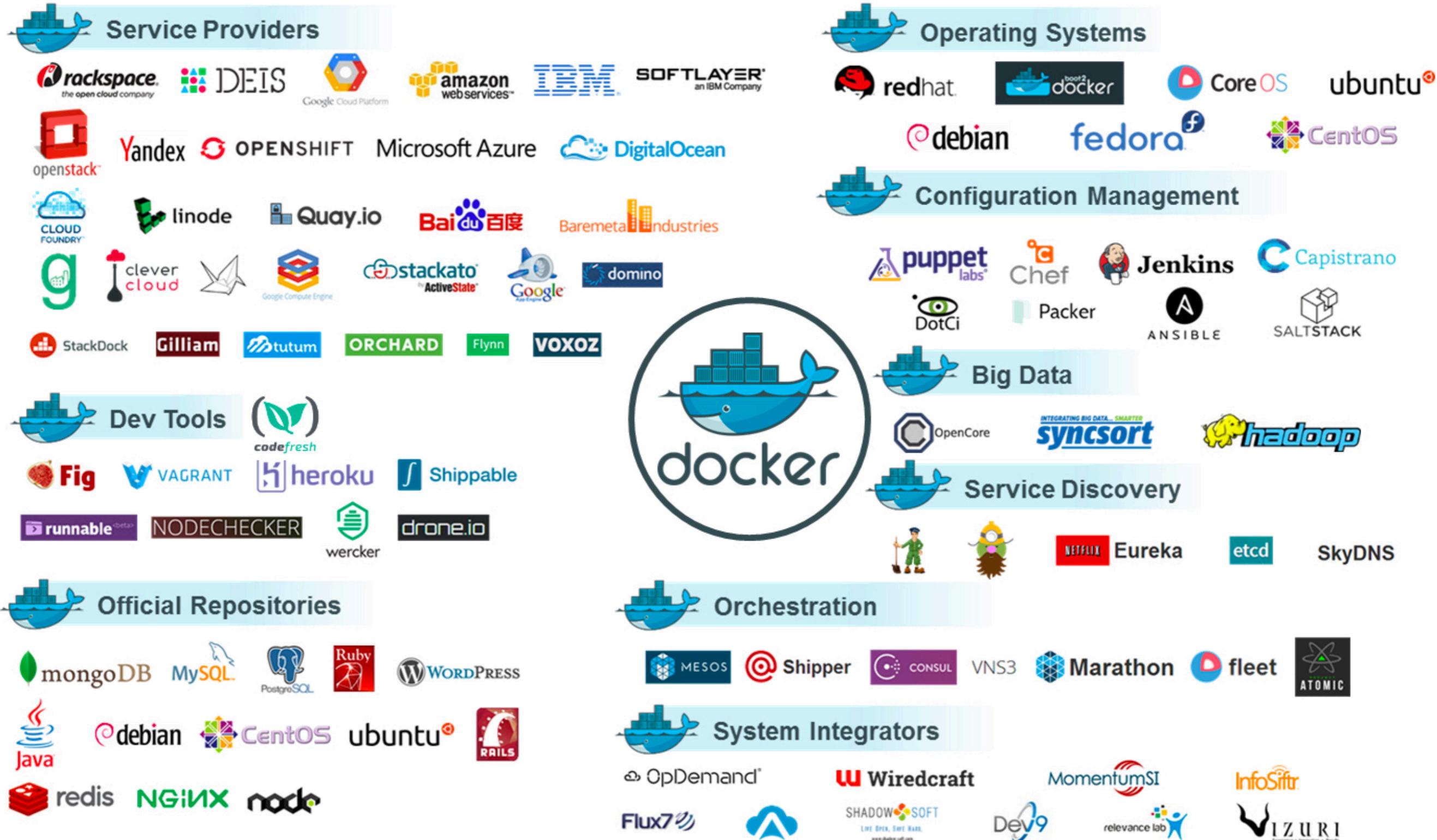
The screenshot shows a GitHub interface with the following details:

- Repository:** docker-library / mysql
- Branch:** master
- Issues:** 17
- Pull requests:** 10
- Security:** Insights
- Watch:** 96
- Star:** 1,404
- Fork:** 1,399
- Latest commit:** ed0e47e on Jul 22
- Commit History:**
  - Dockerfile: Update to 5.6.45-1debian9 (last month)
  - docker-entrypoint.sh: fix silently skipped init scripts (last year)

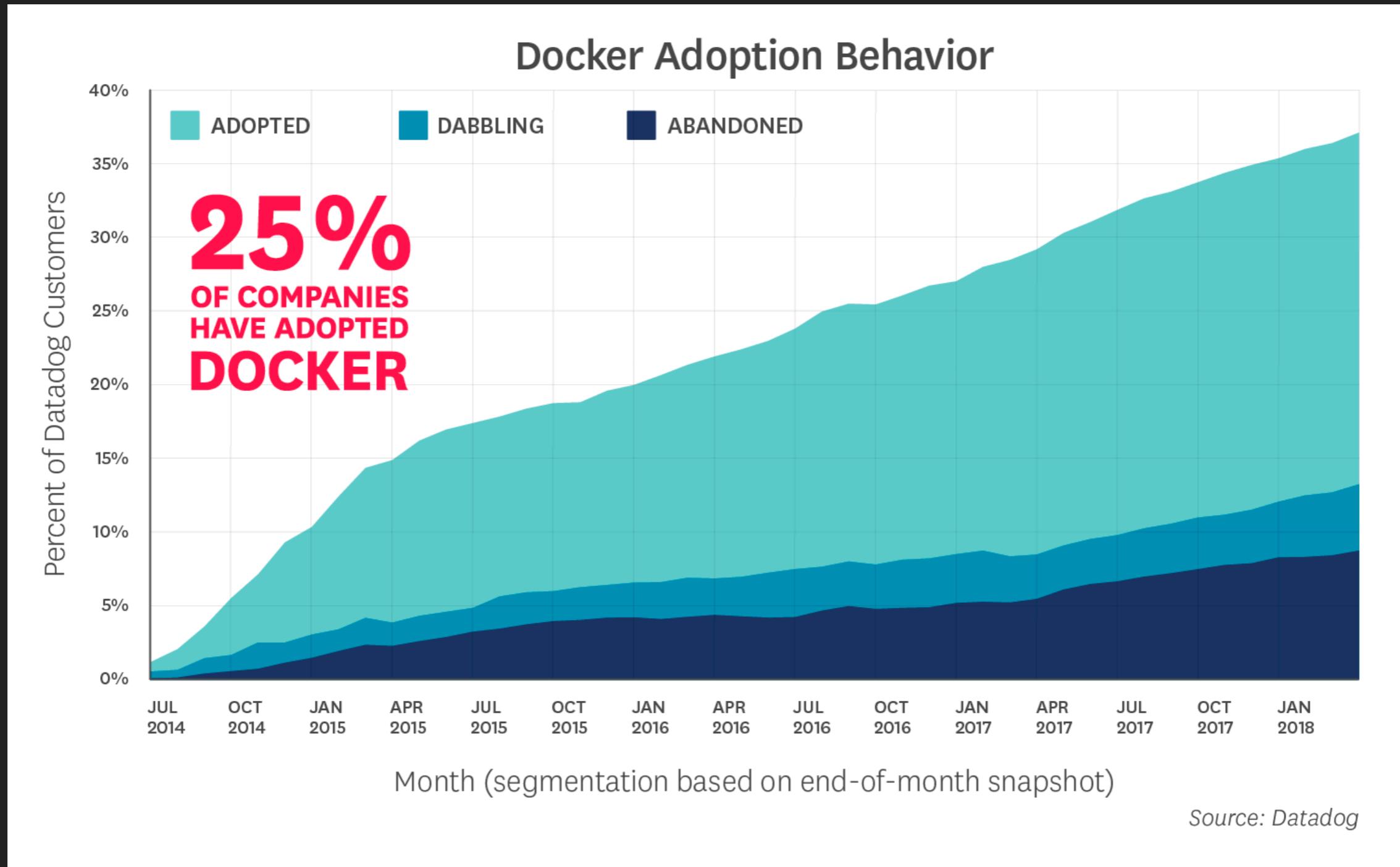
At the bottom, there are links for GitHub services: Contact GitHub, Pricing, API, Training, Blog, and About.

# DOCKER - WHAT'S DOCKER

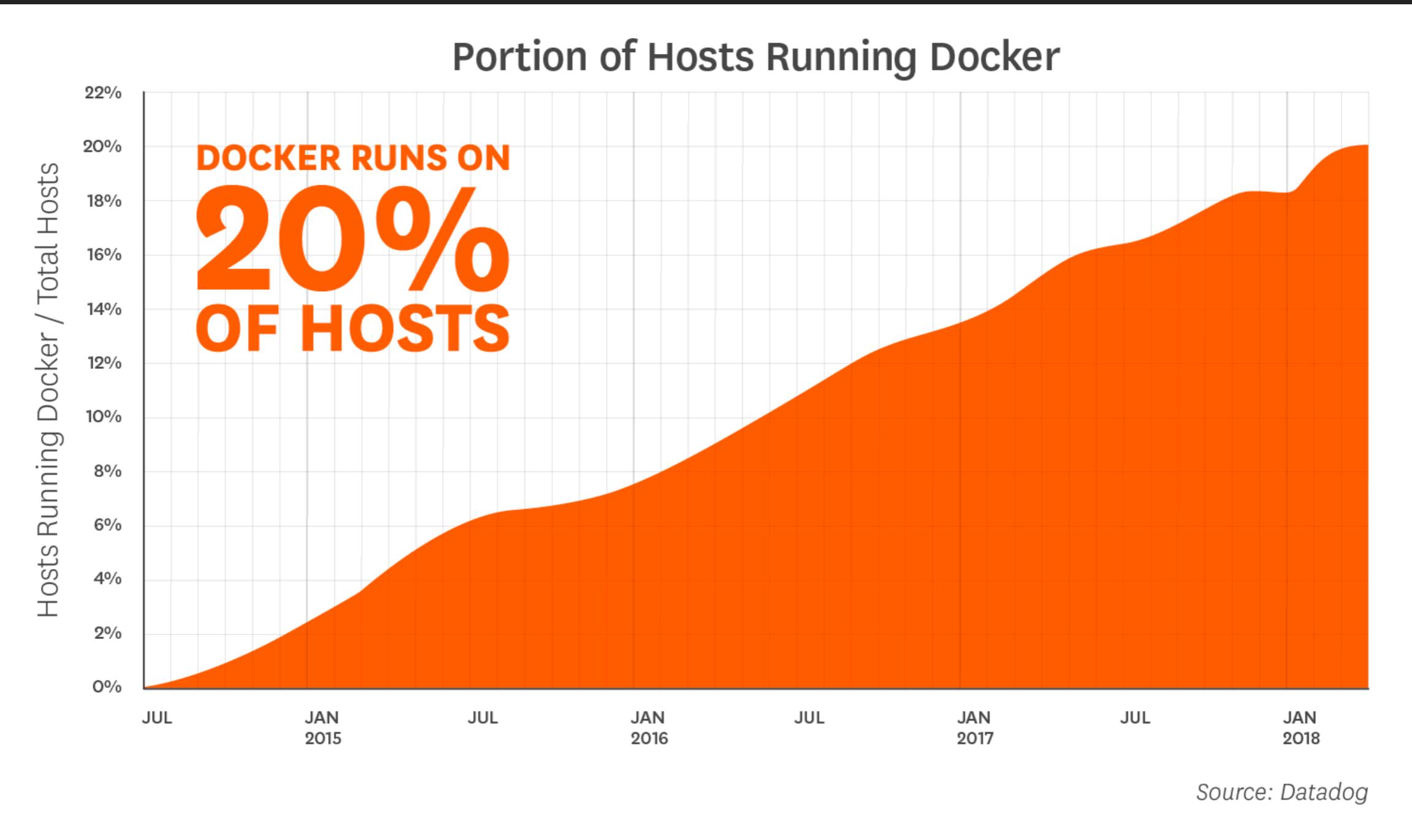
## ECOSYSTEM



# DOCKER-ADOPTION @ DATADOGHQ.COM



# DOCKER-ADOPTION @ DATADOGHQ.COM



## DOCKER-ADOPTION @ DATADOGHQ.COM

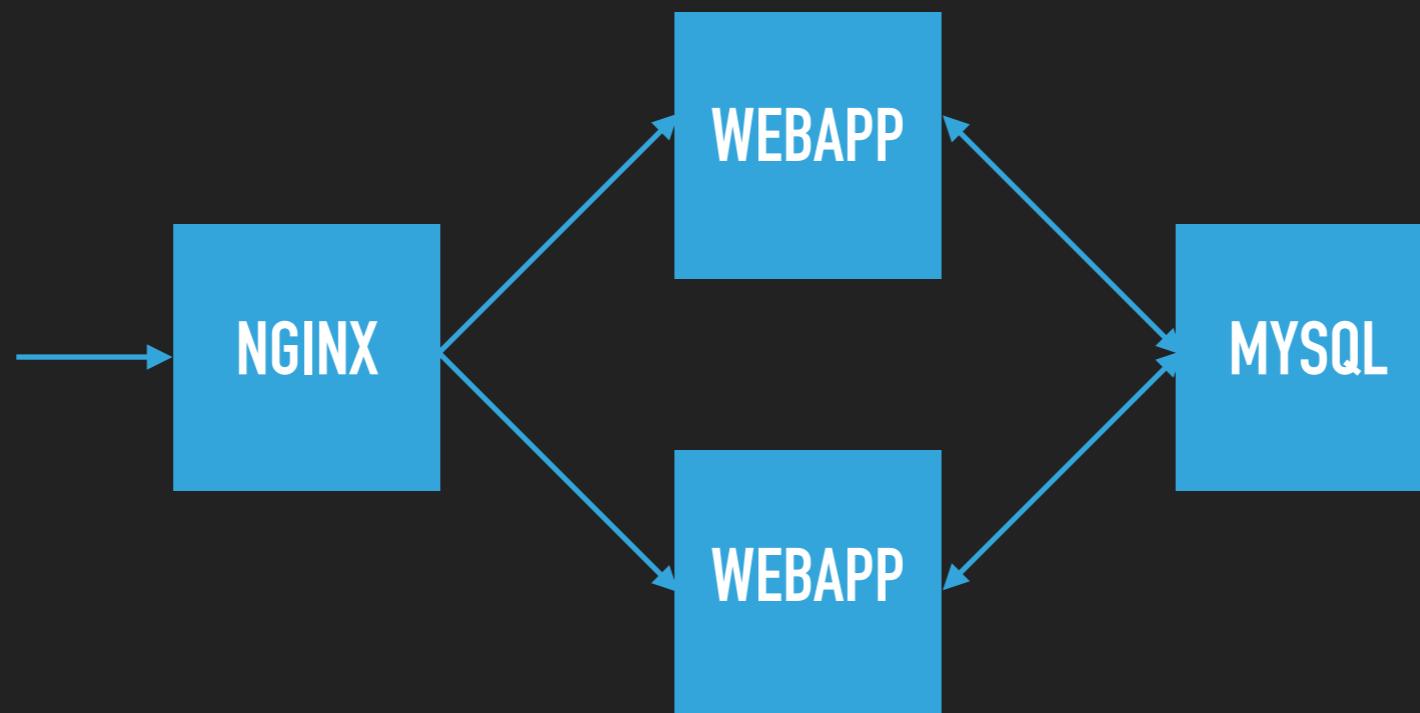
- ▶ <https://www.datadoghq.com/docker-adoption/>

## GIT HUB REPO

► [https://github.com/zenanswer/docker\\_practice](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

## 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/9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849/resolv.conf",
 "HostnamePath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849/hostname",
 "HostsPath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849/hosts",
 "LogPath": "/var/lib/docker/containers/9a5e3439bd8a8a2d14655238a83795cf98207622d5679ba4f10b732010ca0849/log",
 "Name": "/dp_mysql",
 "RestartCount": 0,
 "Driver": "overlay2",
 "Platform": "linux",}
```

## DOCKER CLI

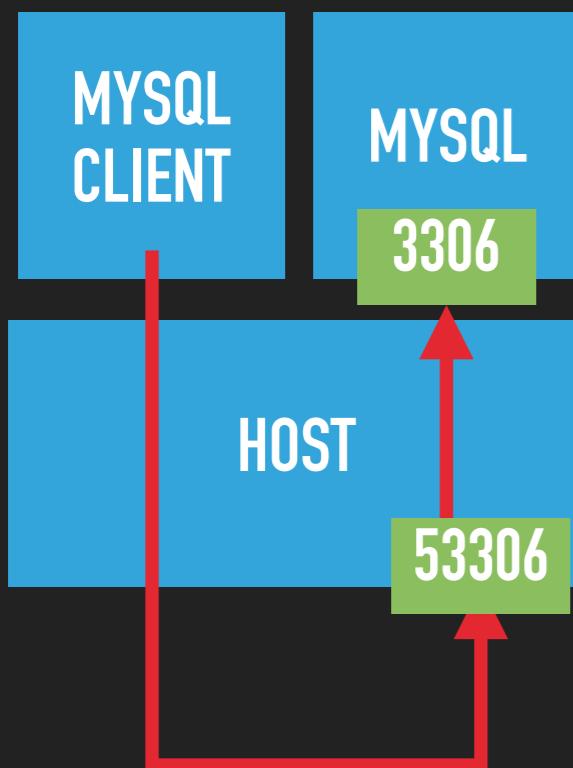
### ▶ MySql DB

#### ▶ 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   0 Aug 14 06:09 .
drwxr-xr-x  1 root ro   0 Aug 14 06:09 ..
drwxr-xr-x  2 root ro   0 Aug 14 06:09 Hello
drwxr-xr-x  2 root ro   0 Aug 14 06:09 world
drwxr-xr-x  2 root ro   0 Aug 14 06:09 .
root@9a5e3439bd8a:/# echo 'Hello world!' > Hello/world
root@9a5e3439bd8a:/# cat Hello/world
Hello world!
root@9a5e3439bd8a:/#
```

## DOCKER CLI

### ▶ MySql Client



```
localhost:mysql xcwang$ cat mysql_cli.sh
docker run -it --rm mysql:5.6 mysql -h192.168.31.221 -P53306 -u$USER -p$PASSWORD
```

```
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$ █
```

## 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
    └── views
        ├── add-player.ejs
        ├── edit-player.ejs
        ├── index.ejs
        └── partials
            └── header.ejs
    └── public
        └── assets
            └── img
    └── run.sh
    └── 7 directories, 13 files
```

```
🚢 Dockerfile ✘
🚢 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
```

## DOCKERFILE

- ▶ WebAPP
- ▶ Dockerfile & docker build

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

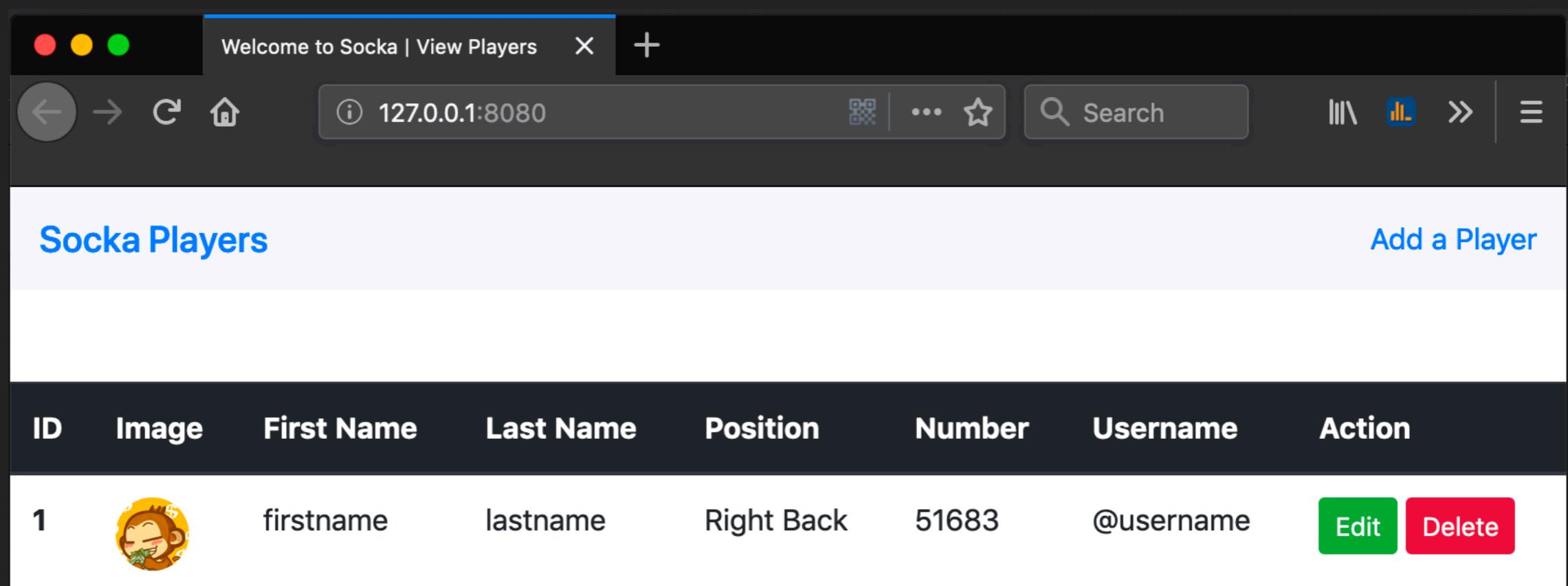
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dp_webapp	latest	9a1989ad8ca2	34 minutes ago	79.3MB
<none>	<none>	245e28adc24a	41 minutes ago	79.3MB
nginx	1.17.3	5a3221f0137b	2 weeks ago	126MB
mysql	5.6	732765f8c7d2	2 weeks ago	257MB
ubuntu	18.04	3556258649b2	5 weeks ago	64.2MB
golang	1.11	ee3ec9ac0398	6 weeks 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-nodejs-and-mysql-19me>
  - ▶ <https://github.com/achowba/node-mysql-crud-app>

## DOCKERCLI

### ▶ NGINX

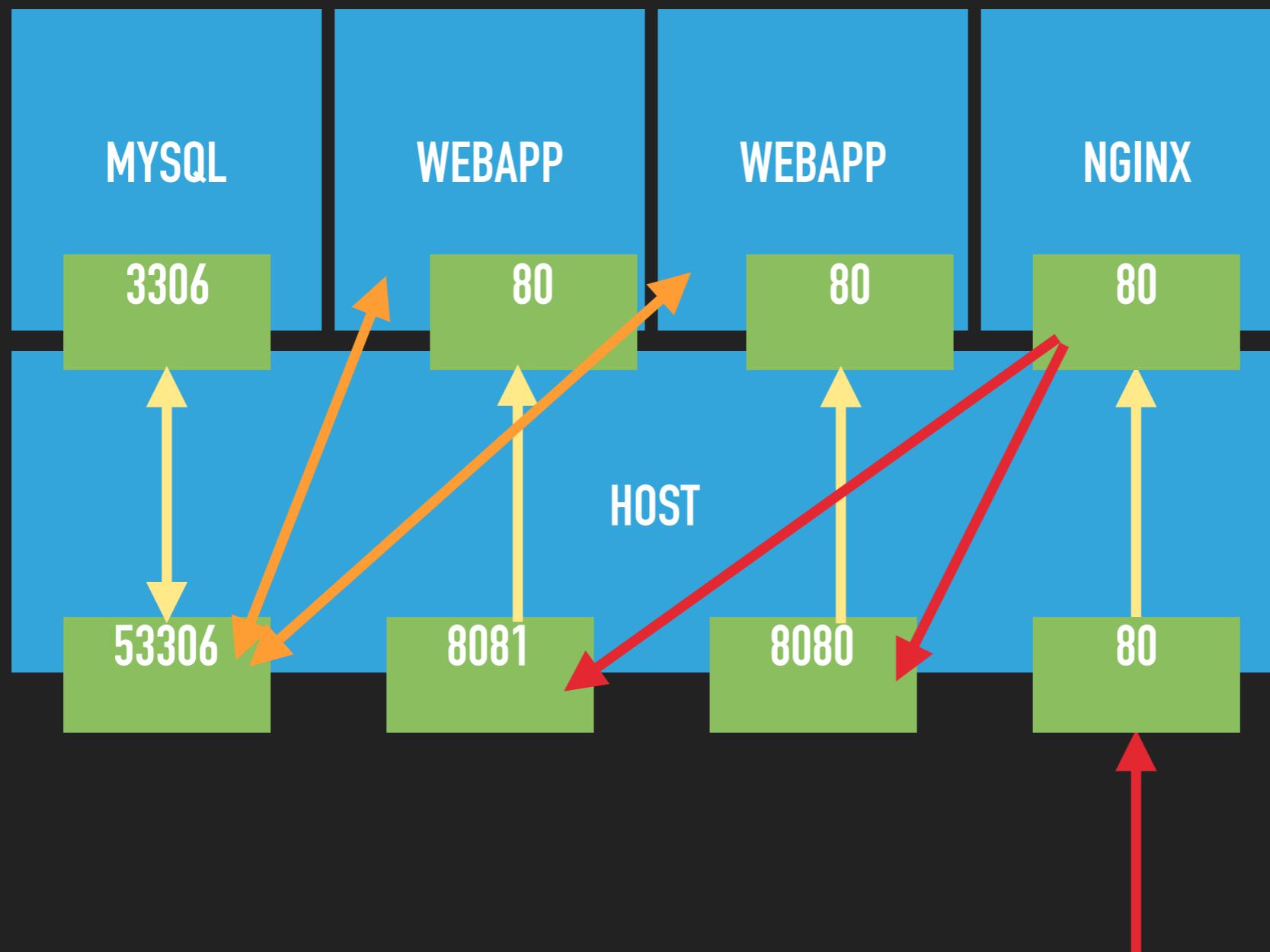
```
localhost:nginx xcwang$ cat nginx.conf
http {
    upstream backend {
        least_conn;
        server 192.168.31.221:8080;
        server 192.168.31.221:8081;
    }

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

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

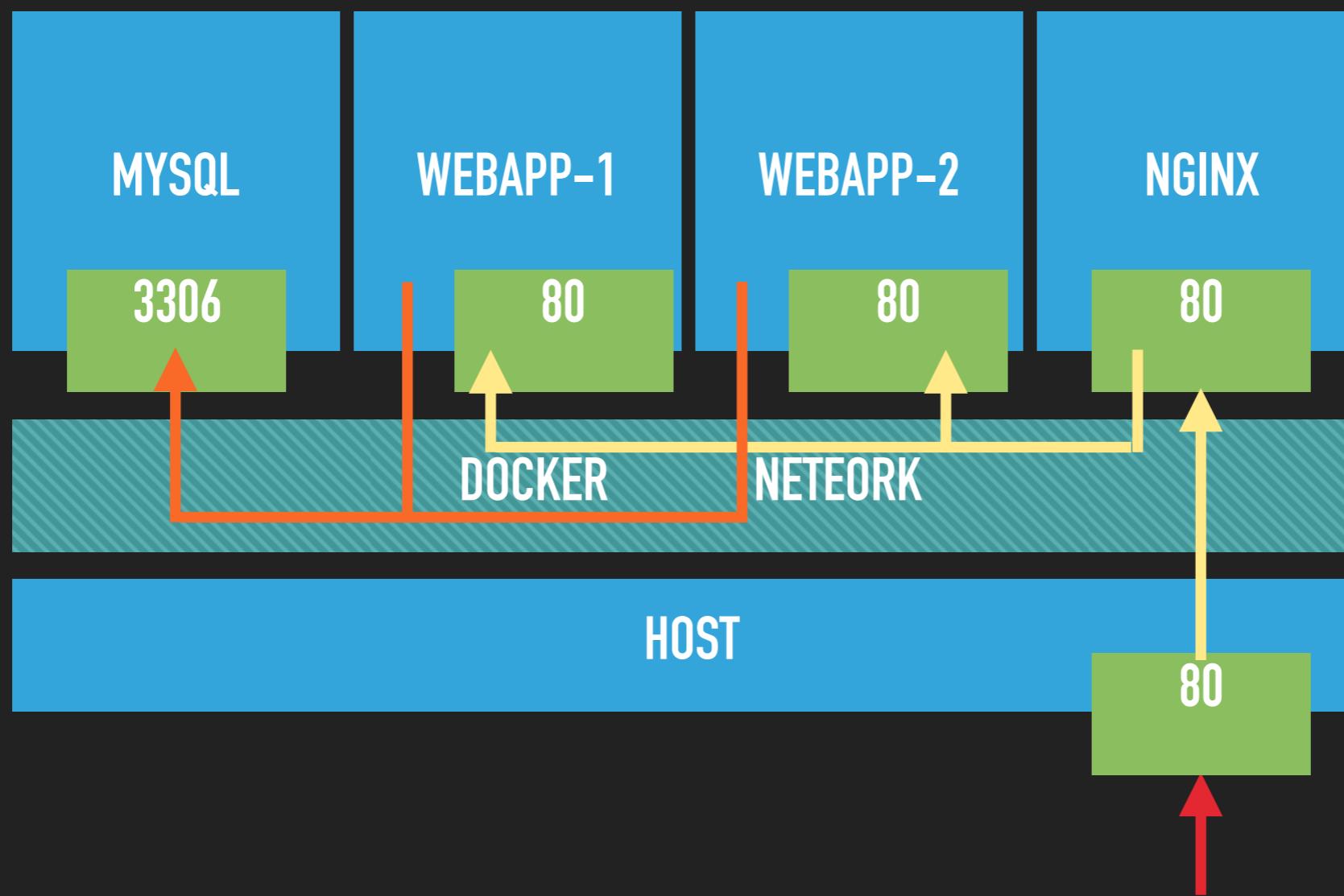
```
docker run \
-v `pwd`/nginx.conf:/etc/nginx/nginx.conf \
-p 80:80 \
nginx:1.17.3
```

# DOCKERCLI



## DOCKER COMPOSE & CLI

- ▶ MySql DB + 2 \* Web APP + Nginx



# DOCKER COMPOSE & CLI

## ► Docker compose file

```
dockers docker-compose.yaml nginx.conf
docker > docker-compose.yaml
You, 6 minutes ago | 1 author (You)
1 http {
2     upstream backend {
3         least_conn;
4         server webapp;
5     }
6
7     server {
8         listen 80;
9         location / {
10            proxy_pass http://backend;
11        }
12    }
13 }
14
15 events {
16     worker_connections 1024; ## Default: 1024
17 }
18
```

```
dockers docker-compose.yaml
dockers docker > docker-compose.yaml > abcversion
You, 2 minutes ago | 1 author (You)
1 version: "3.2" You, 2 minutes ago • docker-compose works an
2 services:
3   mysql:
4     image: mysql:5.6
5     restart: on-failure
6     volumes:
7       - type: bind
8         source: ./data/mysql
9         target: /var/lib/mysql
10      - type: bind
11        source: ../webapp/node-mysql-crud-app/create_table.sql
12        target: /docker-entrypoint-initdb.d/create_table.sql
13 environment:
14   - MYSQL_ROOT_PASSWORD=root
15   - MYSQL_USER=dpuser
16   - MYSQL_PASSWORD=dppassed
17   - MYSQL_DATABASE=dp
18   webapp:
19     image: dp_webapp
20     restart: on-failure
21     volumes:
22       - type: bind
23         source: ./data/webapp
24         target: /node-mysql-crud-app/public/assets/img
25     environment:
26       - DB_URL=mysql://dpuser:dppassed@mysql/dp
27   nginx:
28     image: nginx:1.17.3
29     ports:
30       - "80:80"
31     restart: on-failure
32     volumes:
33       - type: bind
34         source: ./nginx.conf
35         target: /etc/nginx/nginx.conf
36
```

# 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
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$
```

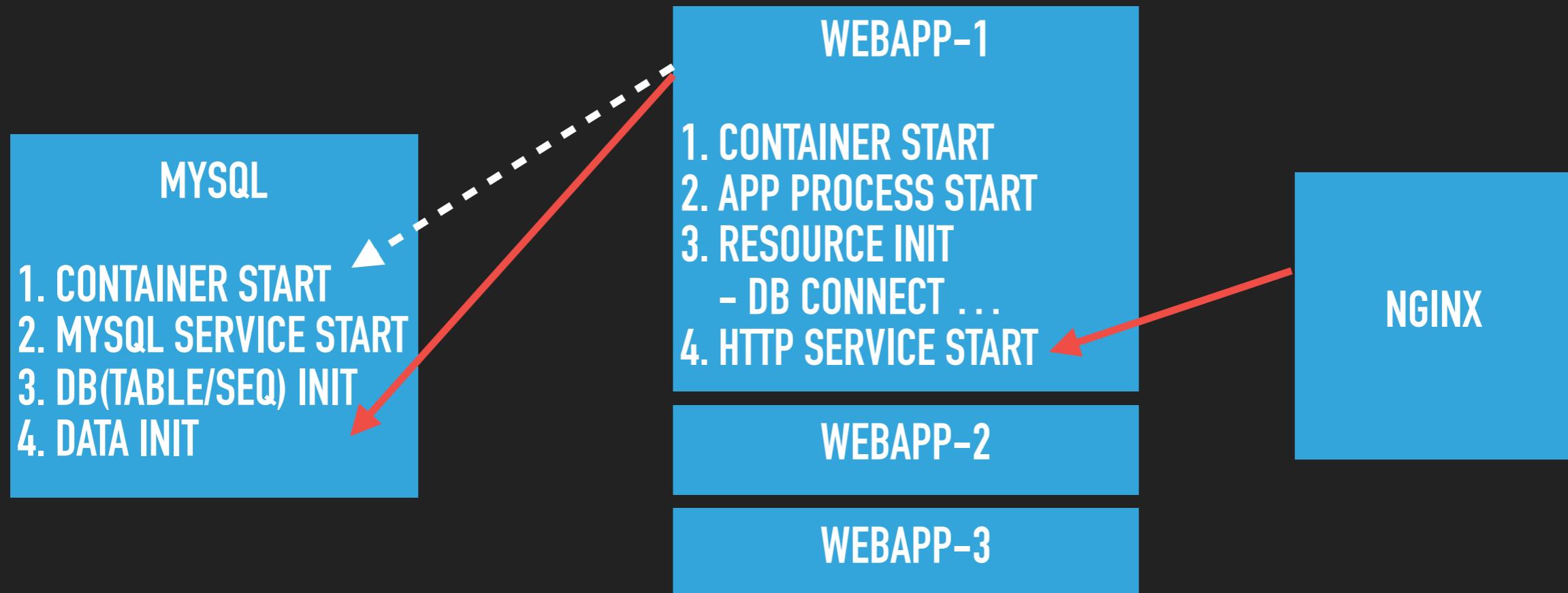
# DOCKER COMPOSE & CLI

## ▶ 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

```
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
```

```
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
    - type: bind
      source: ./checkiftableexists.sh
      target: /checkiftableexists.sh
  environment:
    - MYSQL_ROOT_PASSWORD=root
    - MYSQL_USER=dpuser
    - MYSQL_PASSWORD=dppassed
    - MYSQL_DATABASE=dp
  healthcheck:
    test: ["CMD", "bash", "/checkiftableexists.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

