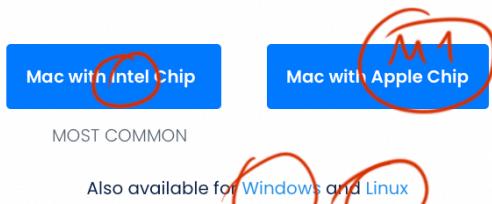


# Docker Desktop

Install Docker Desktop – the fastest way to containerize applications.



```
alex@rr: ~
$ docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers
```

layers



```
8: Pulling from library/mysql
4be31f6562f: Pull complete
96e2eb237a1b: Pull complete
8aa3ac85066b: Pull complete
ac7e524f6c89: Pull complete
f6a88631064f: Pull complete
15bb3ec3ff50: Pull complete
ae65dc337dcb: Pull complete
573c3c7fa18d: Pull complete
9d10771b98b8: Downloading 51.69MB/109.1MB
3d8ef442614b: Download complete
7dc17a6cea26: Download complete
752752efdaea: Download complete
```

Docker — 1. containerization

## • PostgreSQL

```
alex@rr: ~
$ docker pull postgres:9.6
9.6: Pulling from library/postgres
Digest: sha256:cadd35b05cdd56c614ab1f674e63be778e0abdf54e71a7507ff3e28d4902698
Status: Image is up to date for postgres:9.6
docker.io/library/postgres:9.6
alex@rr: ~
```

①



*docker pull postgres*

#2

```
Digest: sha256:2dade3f044f140ec6c07716d34f0b317b98f8e251435abd347951699f7aa3904
Status: Downloaded newer image for mysql:8
docker.io/library/mysql:8
```

```
→ ~ docker pull postgres:9.6
```

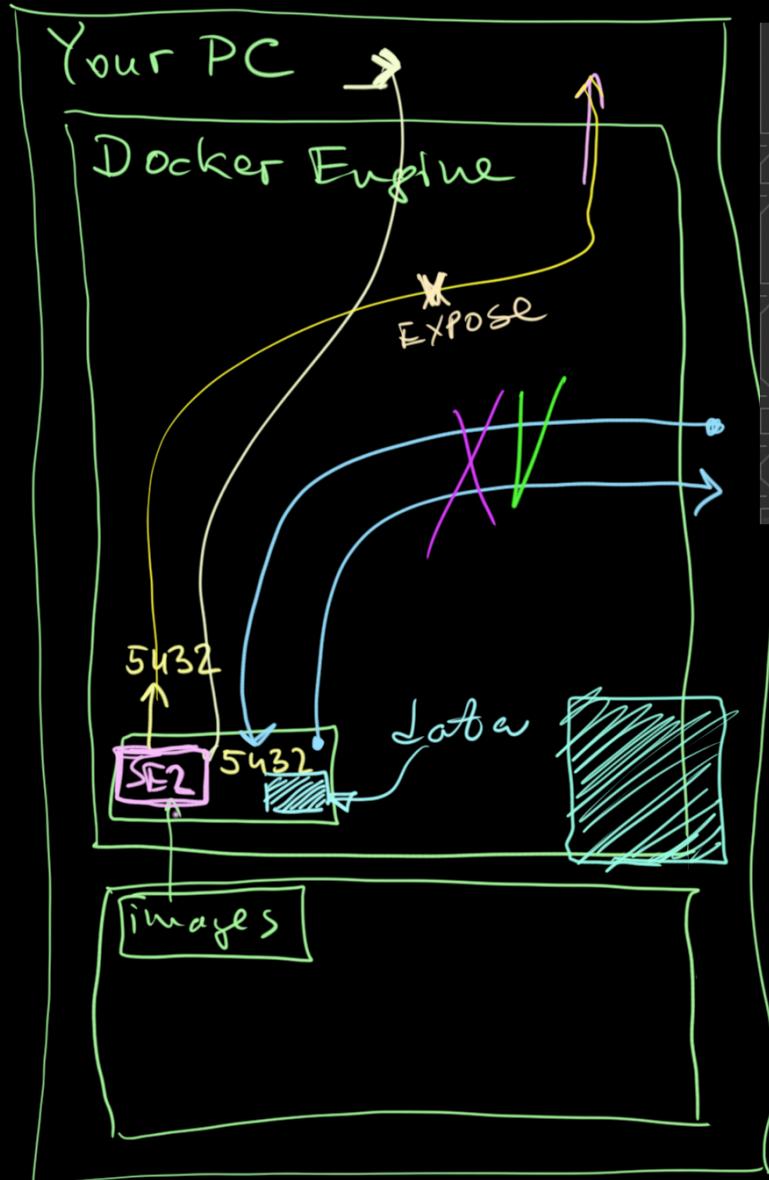
```
9.6: Pulling from library/postgres
```

```
Digest: sha256:cadd35b05cdd56c614ab1f674e63be778e0abdf54e71a7507ff3e28d4902698
Status: Image is up to date for postgres:9.6
docker.io/library/postgres:9.6
```

```
→ ~ docker images
```

| REPOSITORY        | TAG    | IMAGE ID     | CREATED      | SIZE  |
|-------------------|--------|--------------|--------------|-------|
| user-auth-profile | latest | d56cf9ae8b11 | 2 hours ago  | 262MB |
| <none>            | <none> | ba0b27038707 | 3 hours ago  | 226MB |
| scala-http-docker | 0.0.1  | 9269eab64bf8 | 3 hours ago  | 557MB |
| mysql             | 8      | 96d0eae5ed60 | 6 days ago   | 524MB |
| postgres          | 9.6    | 027ccf656dc1 | 2 months ago | 200MB |

#3



```
version: '3.3'
services:
  postgres:
    image: postgres:9.6
    environment:
      POSTGRES_USER: 'postgres'
      POSTGRES_PASSWORD: 'pg123456'
    ports:
      - 5432:5432
```

② docker-compose.yml

port from the postgres cont.

bws port on the HOST machine

Run!

- docker run ...
- docker-compose up

docker-compose.yml

```
→ pg96 git:(master) ✘ docker-compose up
[+] Running 1/0
  ● Container pg_0 Recreated
Attaching to pg_0
pg_0
pg_0 | PostgreSQL Database directory appears to contain a database; Skipping initialization
pg_0 |
pg_0 | LOG:  database system was shut down at 2022-05-04 15:29:18 UTC
pg_0 | LOG:  MultiXact member wraparound protections are now enabled
pg_0 | LOG:  autovacuum launcher started
pg_0 | LOG:  database system is ready to accept connections
```

(3)

| CONTAINER ID | IMAGE        | COMMAND                  | CREATED            | STATUS            | PORTS                  | NAMES |
|--------------|--------------|--------------------------|--------------------|-------------------|------------------------|-------|
| 1f02f858b4e2 | postgres:9.6 | "docker-entrypoint.s..." | About a minute ago | Up About a minute | 0.0.0.0:5432->5432/tcp | pg_0  |

Jump into container

```
→ ~ docker exec -it 1f02f858b4e2 bash
root@1f02f858b4e2:/# su postgres
postgres@1f02f858b4e2:/$ createdb se2
postgres@1f02f858b4e2:/$
```

postgres command

(4)

#5

Host: localhost 127.0.0.1

Port: 5432

Authentication: User & Password

User: postgres

Password: <hidden>

Save: Until restart

Database: se2

URL: jdbc:postgresql://localhost:5432/se2  
Overrides settings above

```
version: '3.3'
services:
  postgres:
    image: postgres:9.6
    environment:
      POSTGRES_USER: 'postgres'
      POSTGRES_PASSWORD: 'pg123456'
    ports:
      - 5432:5432
```

```
→ ~ docker exec -it 1f02f858b4e2 bash
root@1f02f858b4e2:/# su postgres
postgres@1f02f858b4e2:/$ createdb se2
postgres@1f02f858b4e2:/$
```

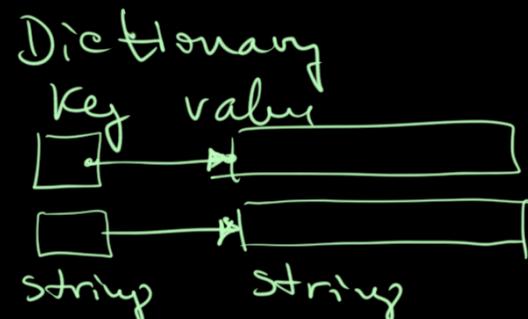
configuration As A Code CAAc  
 Infrastructure as a code IAAc



- MySQL disk
- PostgreSQL
- Oracle SQL
- Redis mem
- Kafka mem
- ...

JAVA  
HASH MAP ?

Queues, Pipes



Redis =  
HashMap<String, String>

you can put your own code into docker containers.

→ ~ docker images

| REPOSITORY        | TAG    | IMAGE ID     | CREATED            | SIZE  |
|-------------------|--------|--------------|--------------------|-------|
| scala-http-docker | 0.0.1  | 157fd42cd4e2 | About a minute ago | 557MB |
| user-auth-profile | latest | d56cf9ae8b11 | 3 hours ago        | 262MB |
| <none>            | <none> | ba0b27038707 | 4 hours ago        | 226MB |
| <none>            | <none> | 9269eab64bf8 | 4 hours ago        | 557MB |
| mysql             | 8      | 96d0ea5ed60  | 6 days ago         | 524MB |
| postgres          | 9.6    | 027ccf656dc1 | 2 months ago       | 200MB |

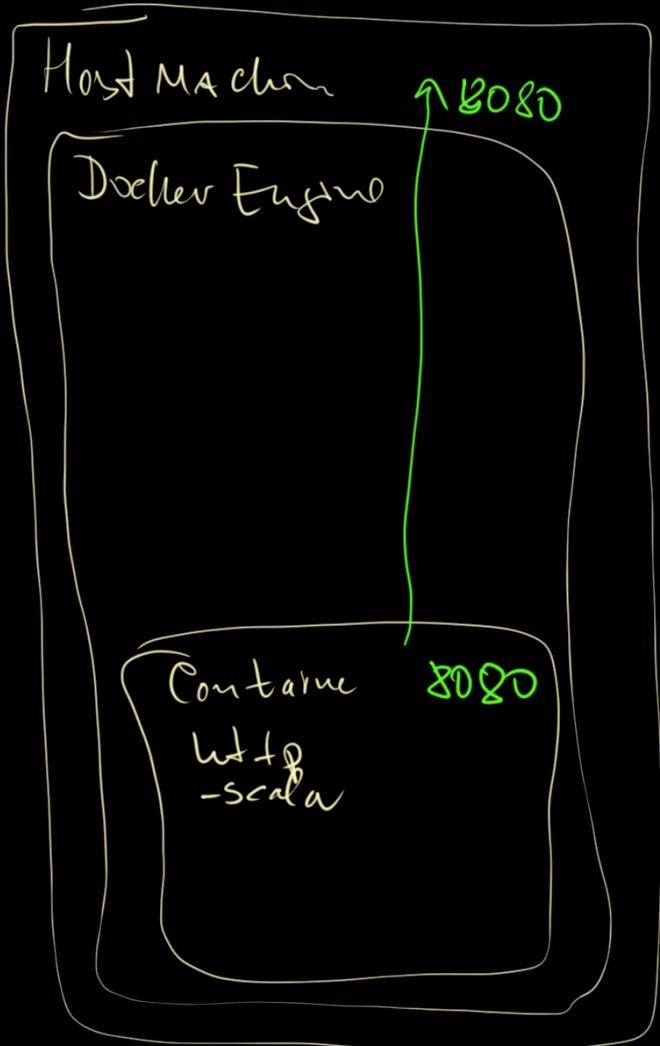
→ scala-http-docker git:(master) ✘ cat docker-compose.yml

```

version: '3.3'
services:
  simple-web:
    image: scala-http-docker:0.0.1
    ports:
      - '127.0.0.1:8081:8080'
  
```

override def run: IO[Unit] = BlazeServerBuilder[IO]
 .bindHttp(port = 8080, host = "0.0.0.0")
 .withHttpApp(routes.orNotFound)
 .serve



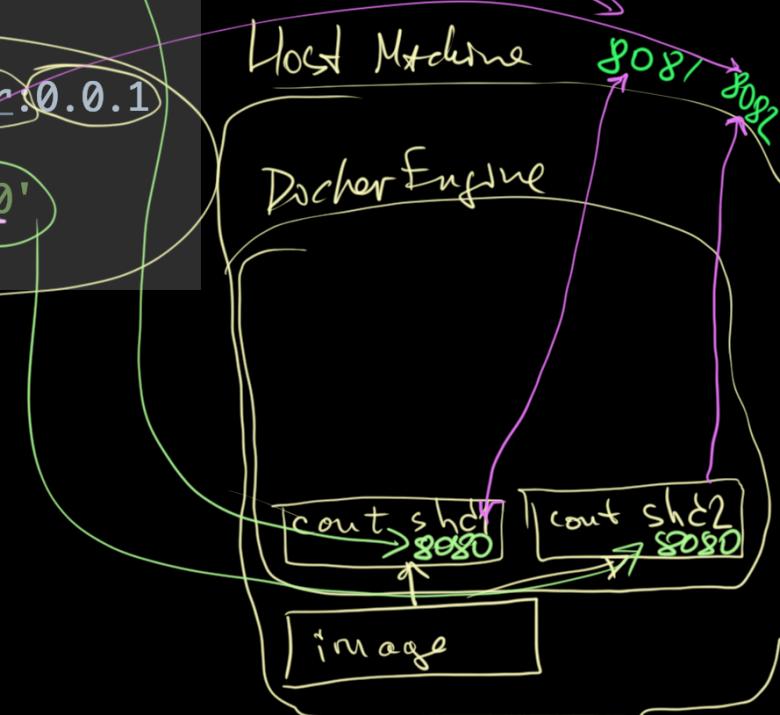


```
version: '3.3'

services:
  simple1:
    image: scala-http-docker:0.0.1
    ports:
      - '127.0.0.1:8081:8080'
    container_name: shd_1

  simple2:
    image: scala-http-docker:0.0.1
    ports:
      - '127.0.0.1:8082:8080'
    container_name: shd_2
```

but there is the way  
to unite them  
into one network



```
val routes = HttpRoutes.of[IO] {
    case GET -> Root
        => IO(println("=>")) >> Ok("It works")
```

→ scala-http-docker git:(master) ✘ docker-compose up

→ ~ curl http://localhost:8081  
sendup HTTP request

"It works"%

→ ~ curl http://localhost:8082

"It works"%

→ ~ curl http://localhost:8081  
"It works"%  
→ ~ curl http://localhost:8081  
"It works"%  
→ ~ curl http://localhost:8081  
"It works"%  
→ ~ curl http://localhost:8081  
"It works"%  
→ ~ curl http://localhost:8081  
"It works"%  
→ ~ curl http://localhost:8081  
"It works"%  
→ ~

=> IO(println("=>")) >> Ok("It works")

shd\_1  
shd\_2

shd\_1  
shd\_2  
shd\_1  
shd\_1  
shd\_1  
shd\_1  
shd\_1

>>  
>>

>>  
>>  
>>  
>>  
>>  
>>

```

<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.22</version>
  <scope>provided</scope>
</dependency>

```

Just a library share

```

container_name: pg_0

→ pg96 git:(master) ✘ docker-compose up
[+] Running 1/0
  ⚡ Container pg_0 Recreated
  ↳ Attaching to pg_0
pg_0 | PostgreSQL Database directory appears to c
pg_0 | LOG:  database system was shut down at 202
pg_0 | LOG:  MultiXact member wraparound protecti
pg_0 | LOG:  autovacuum launcher started
pg_0 | LOG:  database system is ready to accept c
pg_0 | WARNING:  could not open statistics file

```

```

→ scala-http-docker git:(master) ✘ docker-compose up
WARN[0000] Found orphan containers ([shd_2 shd_1]) for this
          with the --remove-orphans flag to clean it up.
[+] Running 1/0
  ⚡ Container scala-http-docker-simple-web-1 Recreated
    0.1s
  ↳ Attaching to scala-http-docker-simple-web-1, shd_1, shd_2
scala-http-docker-simple-web-1 | SLF4J: Failed to load clas
scala-http-docker-simple-web-1 | SLF4J: Defaulting to no-op
scala-http-docker-simple-web-1 | SLF4J: See http://www.slf4
scala-http-docker-simple-web-1 | SLF4J: Failed to load clas
scala-http-docker-simple-web-1 | SLF4J: Defaulting to no-op
scala-http-docker-simple-web-1 | SLF4J: See http://www.slf4
scala-http-docker-simple-web-1 | SLF4J: Failed to load clas
scala-http-docker-simple-web-1 | SLF4J: Defaulting to no-op
scala-http-docker-simple-web-1 | SLF4J: See http://www.slf4

```

```

simple1:
  image: scala-http-docker:0.0.1
  ports:
    - '127.0.0.1:8081:8080'
  container_name: shd_1

```

Just an whole app share

```

services:
  simple1:
    image: scala-http-docker:0.0.1
    ports:
      - '127.0.0.1:8081:8080'
    container_name: shd_1

```

```

.e2:
  image: scala-http-docker:0.0.1
  ports:
    - '127.0.0.1:8082:8080'
  container_name: shd_2

```

#9

→ ~ docker images ↗ library

| REPOSITORY        | TAG    | IMAGE ID     | CREATED        | SIZE  |
|-------------------|--------|--------------|----------------|-------|
| scala-http-docker | 0.0.1  | 157fd42cd4e2 | 31 minutes ago | 557MB |
| user-auth-profile | latest | d56cf9ae8b11 | 4 hours ago    | 262MB |
| <none>            | <none> | ba0b27038707 | 4 hours ago    | 226MB |
| <none>            | <none> | 9269eab64bf8 | 5 hours ago    | 557MB |
| mysql             | 8      | 96d0eae5ed60 | 6 days ago     | 524MB |
| postgres          | 9.6    | 027ccf656dc1 | 2 months ago   | 200MB |

→ containers & runnable artifact

| CONTAINER ID | IMAGE                   | COMMAND                  | CREATED           | STATUS                 | PORTS                    | NAMES                          |
|--------------|-------------------------|--------------------------|-------------------|------------------------|--------------------------|--------------------------------|
| b15fe0e376bf | scala-http-docker:0.0.1 | "/opt/docker/bin/sca..." | 23 minutes ago    | Up 23 minutes          | 0.0.0.0:8080->8080/tcp   | scala-http-docker-simple-web-1 |
| 1f02f858b4e2 | postgres:9.6            | "docker-entrypoint.s..." | About an hour ago | Up About an hour       | 0.0.0.0:5432->5432/tcp   | pg_0                           |
| 0ba5692e6d5f | postgres:9.6            | "docker-entrypoint.s..." | 2 hours ago       | Exited (1) 2 hours ago |                          | wonderful_morse                |
| 32720de67ece | 9269eab64bf8            | "/opt/docker/bin/sca..." | 2 hours ago       | Up 23 minutes          | 127.0.0.1:8082->8080/tcp | shd_2                          |
| f2408140d58f | 9269eab64bf8            | "/opt/docker/bin/sca..." | 2 hours ago       | Up 23 minutes          | 127.0.0.1:8081->8080/tcp | shd_1                          |

you can start a container → docker run...  
 you can stop a container → docker stop...  
 docker-compose up  
 docker compose s top

```
shd_1
shd_1
^C[+] Running 3/3
  # Container shd_1           Stopped
  # Container scala-http-docker-simple-web-1 Stopped
  # Container shd_2           Stopped
canceled
→ scala-http-docker git:(master) ✘
```

↑ C

→ scala-http-docker git:(master) x docker ps -a → all with status

| CONTAINER ID | IMAGE                   | COMMAND                  | CREATED        | STATUS                          | PORTS | NAMES |
|--------------|-------------------------|--------------------------|----------------|---------------------------------|-------|-------|
| b15fe0e376bf | scala-http-docker:0.0.1 | "/opt/docker/bin/sca..." | 30 minutes ago | Exited (143) About a minute ago |       |       |
| 1f02f858b4e2 | postgres:9.6            | "docker-entrypoint.s..." | 2 hours ago    | Exited (0) 34 seconds ago       |       |       |
| 0ba5692e6d5f | postgres:9.6            | "docker-entrypoint.s..." | 2 hours ago    | Exited (1) 2 hours ago          |       |       |
| 32720de67ece | 9269eab64bf8            | "/opt/docker/bin/sca..." | 3 hours ago    | Exited (143) About a minute ago |       |       |
| f2408140d58f | 9269eab64bf8            | "/opt/docker/bin/sca..." | 3 hours ago    | Exited (143) About a minute ago |       |       |

→ scala-http-docker git:(master) x docker ps → only running

| CONTAINER ID | IMAGE | COMMAND | CREATED | STATUS | PORTS | NAMES |
|--------------|-------|---------|---------|--------|-------|-------|
|              |       |         |         |        |       |       |

→ docker-compose up

container\_name: pg\_0

→ docker run postgres:9.6  
--name my-pg → docker ps

- No way to run the image
- every time new container is created

→ docker push

```
version: '3.3'
```

**services:**
**postgres:**

```
  image: postgres:9.6
  restart: unless-stopped
```

**environment:**

```
  POSTGRES_USER: 'postgres'
  POSTGRES_PASSWORD: 'pg123456'
```

**ports:**

```
- '5432:5432'
```

**volumes:**

```
- ../../../.ro:/ro:ro
- ../../../.rw:/rw
- ../../../.Downloads:/dl:ro
- ../../../.data/pg96:/var/lib/postgresql/data/
```

```
container_name: pg_0
```

this is path  
on the host  
machine

this is path  
in the container

```
root@1f02f858b4e2:/rw# echo "hello from Postgres container" > hello.txt
```

container

```
→ ~ cd ~/Downloads
→ Downloads ls -la
total 1182848
drwxr-xr-x 7 alexr staff 224 Apr 25 10:15 !to_cold
drwx-----@ 13 alexr staff 416 May 4 17:10 .
drwxr-xr-x+ 73 alexr staff 2336 May 4 20:31 ..
-rw-r--r--@ 1 alexr staff 14340 May 3 15:01 .DS_Store
-rw-r--r--@ 1 alexr staff 603740463 May 3 15:00 Docker.dmg
drwxr-xr-x 11 alexr staff 352 May 2 12:27 -
drwxr-xr-x 1285 alexr staff 41120 Apr 30 19:55 _books
drwxr-xr-x 12 alexr staff 384 Apr 27 12:01 _war_images
-rw-r--r--@ 1 alexr staff 1135685 May 3 16:58 api-account-model_2.13-0.2.2-javadoc.jar
drwxr-xr-x@ 15 alexr staff 480 Apr 28 07:48 hospitalization
-rw-r--r--@ 1 alexr staff 0 May 3 17:20 jwt-middleware_2.13-0.2.3.pom
drwxr-xr-x@ 5 alexr staff 160 Apr 27 17:45 templates
drwxr-xr-x@ 7 alexr staff 224 Mar 14 12:04 to SELL
```

```
→ ~ docker exec -it pg_0 bash
```

```
root@1f02f858b4e2:/# cd /dl
root@1f02f858b4e2:/dl# ls -la
total 591428
drwxr-xr-x 11 root root 352 May 2 09:27 -
drwx----- 13 root root 416 May 4 14:10 .
drwxr-xr-x 1 root root 4096 May 4 15:36 ..
-rw-r--r-- 1 root root 1135685 May 3 13:58 api-account-model_2.13-0.2.2-javadoc.jar
drwxr-xr-x 1285 root root 41120 Apr 30 16:55 _books
-rw-r--r-- 1 root root 603740463 May 3 12:00 Docker.dmg
-rw-r--r-- 1 root root 14340 May 3 12:01 .DS_Store
drwxr-xr-x 15 root root 480 Apr 28 04:48 hospitalization
-rw-r--r-- 1 root root 0 May 3 14:20 jwt-middleware_2.13-0.2.3.pom
drwxr-xr-x 5 root root 160 Apr 27 14:45 templates
drwxr-xr-x 7 root root 224 Apr 25 07:15 !to_cold
drwxr-xr-x 7 root root 224 Mar 14 10:04 to SELL
drwxr-xr-x 12 root root 384 Apr 27 09:01 _war_images
root@1f02f858b4e2:/dl#
```

host machine

container

the same

we  
can  
share

```
→ ~ cd ~/rw
→ rw ls -al
total 8
drwxr-xr-x 3 alexr staff 96 May 4 20:35 .
drwxr-xr-x+ 73 alexr staff 2336 May 4 20:36 ..
-rw-r--r-- 1 alexr staff 30 May 4 20:35 hello.txt
→ rw cat hello.txt
hello from Postgres container
→ rw
```

host machine

**Dockerfile**

```

FROM openjdk:8
WORKDIR /opt/docker
COPY 2 /opt /2/opt
COPY 4 /opt /4/opt
EXPOSE 8080
ENTRYPOINT ["./scala-http-docker"]

```

*base Linux + JDK 8*

*creating folder in our new container*

*allow to expose port*

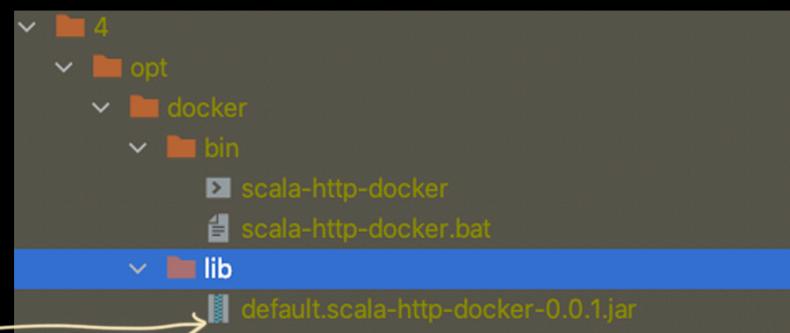
*how to start our app*

*java -jar app.jar*

*docker build => Docker image*

*our code*

hub.docker.com  
~ Maven repo  
~ NPM repo



```
git@github.com:alexr007/scala-http-docker.git
```

```
https://github.com/alexr007/containers.git
```