Actionable Docker

(https://projects.100xdevs.com/tracks/docker-easy/docker-1)

Actionable Docker

Using Docker to run packages locally

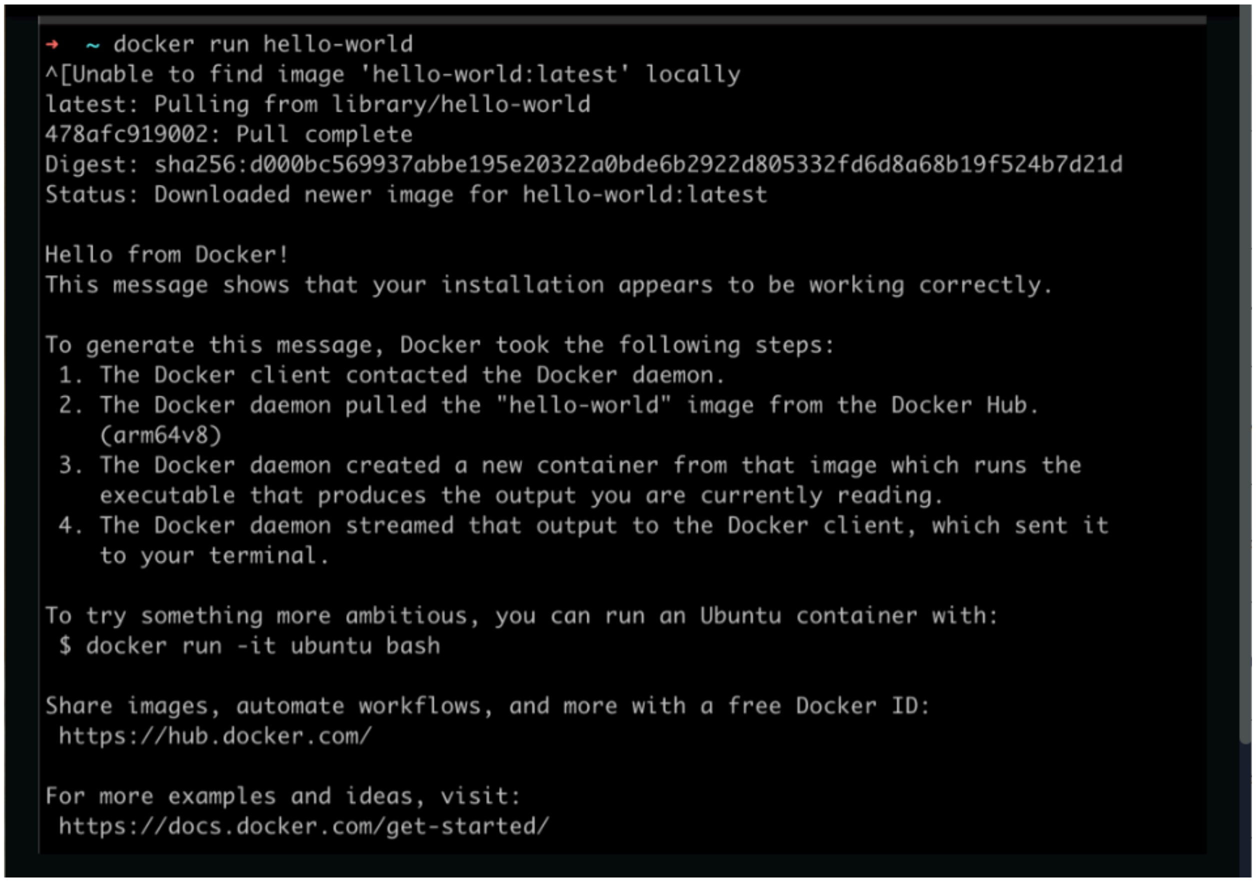
Installing Docker

Docker GUI is the easiest way to get off the ground.

You can find instructions to install docker on

https://docs.docker.com/engine/install/

At the end of the installation, you need to make sure you're able to run the following command -



```
➜  ~ docker run hello-world
^[Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
478afc919002: Pull complete
Digest: sha256:d000bc569937abbe195e20322a0bde6b2922d805332fd6d8a68b19f524b7d21d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (arm64v8)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

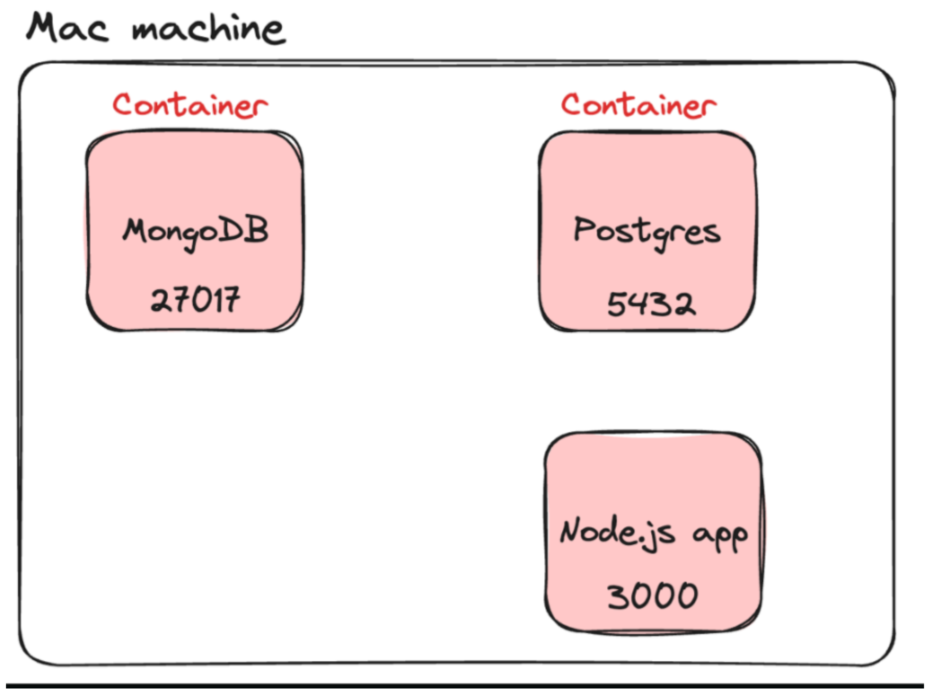
What are we using docker for?

Docker let’s you do a lot of things.

It let’s you containerise your applications.

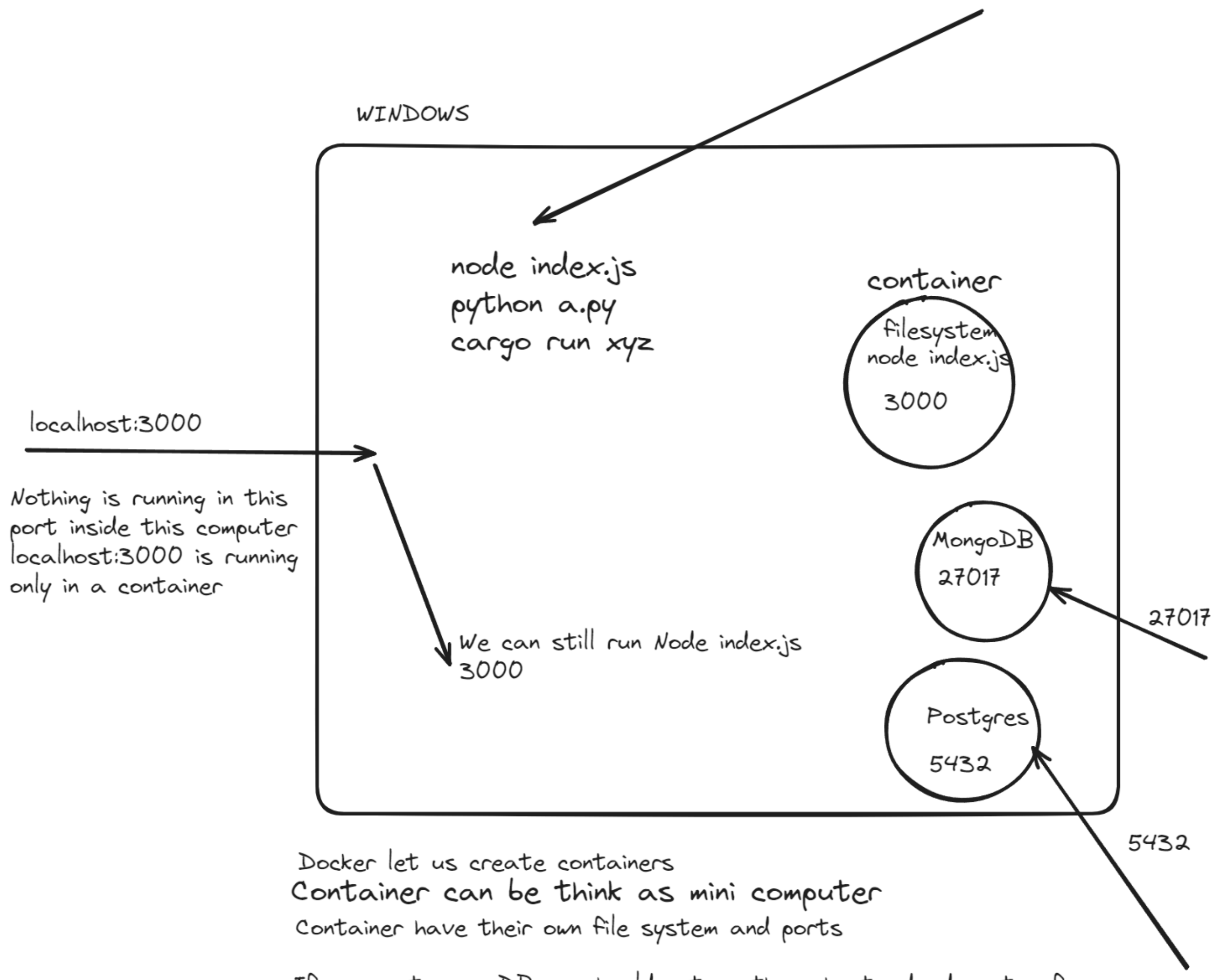
It let’s you run other people’s code + packages in your machine.

It let’s you run common software packages inside a container (For eg - Mongo, Postgres etc)



Containers:---

We can divert the localhost:3000 from outside to the container we want.



Nothing is running in this port inside this computer localhost:3000 is running only in a container

We can still run Node index.js
3000

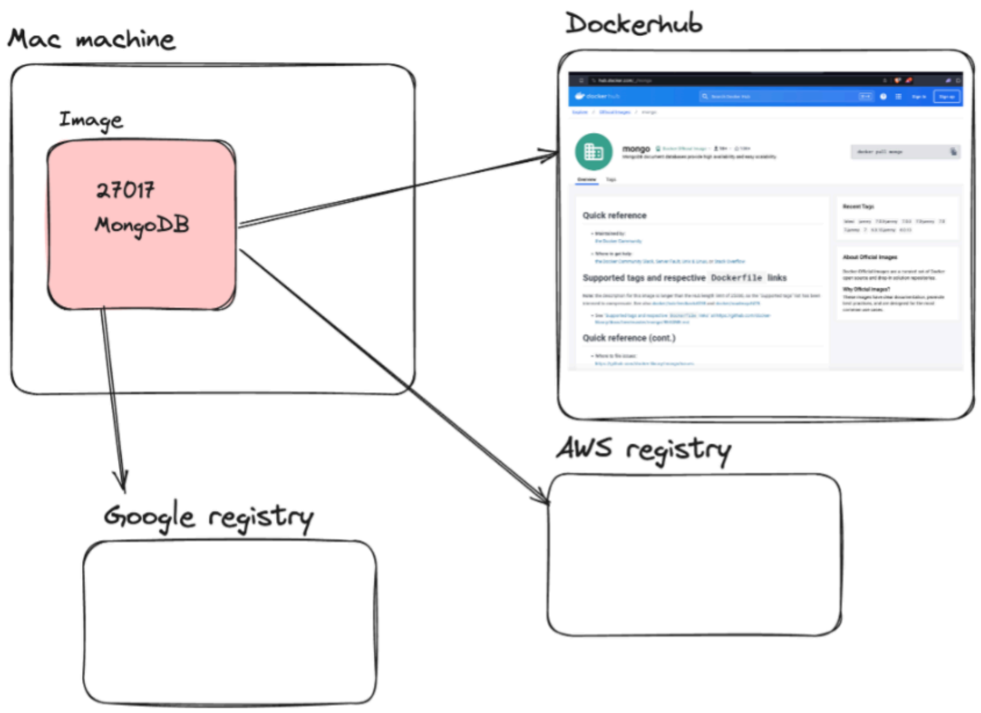
Docker let us create containers
Container can be think as mini computer
Container have their own file system and ports

If we want mongoDB we should not go through standard route of downloading it
We should start a container where we run mongoDB
And we can also start a container where we run postGres

Where can we get packages from??

Just like you can push your code to Github/Gitlab.

You can push **images** to docker registries



What is **images?**

Whoever want other people that can run like mongoDB etc first create an **Image** .

It contains everything mongoDB require and it is packaged together

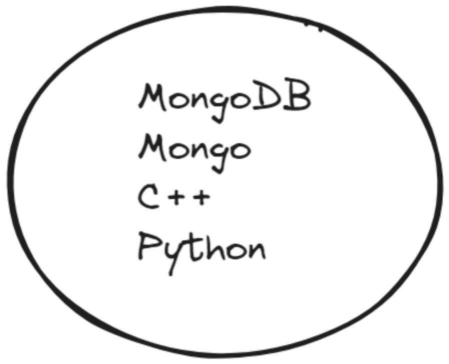


Image is lot of code filesystem , dependencies needed to start MongoDB locally

We can get them from **docker regesteries**

Docker hub mongo

A container is nothing but an image in execution

Common commands to know

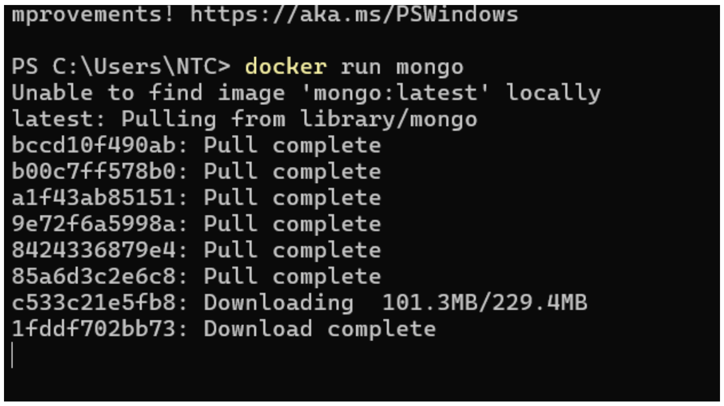
1. docker run
2. docker ps
3. docker kill

Running an image

1. Running a simple image

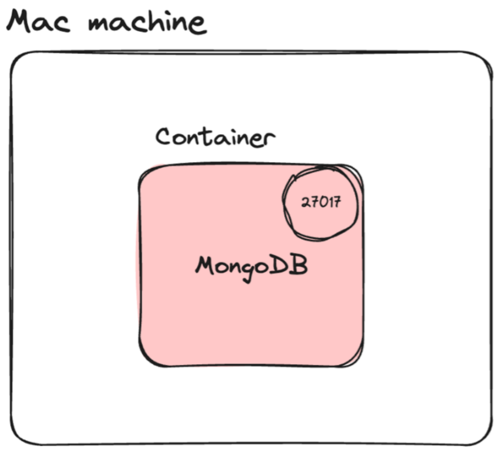
Let’s say we want to run mongoDB locally

docker run mongo

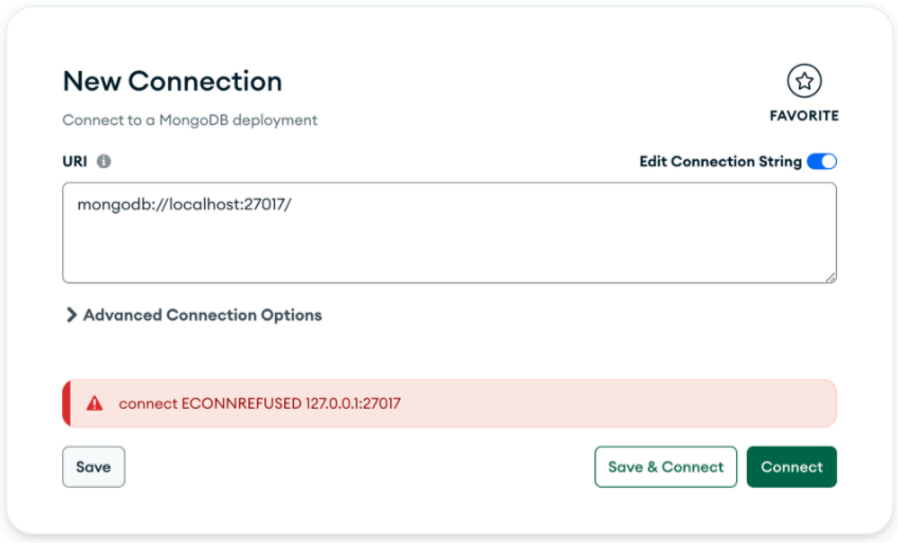


```
mprovements! https://aka.ms/PSWindows

PS C:\Users\NTC> docker run mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
bccd10f490ab: Pull complete
b00c7ff578b0: Pull complete
a1f43ab85151: Pull complete
9e72f6a5998a: Pull complete
8424336879e4: Pull complete
85a6d3c2e6c8: Pull complete
c533c21e5fb8: Downloading  101.3MB/229.4MB
1fddf702bb73: Download complete
```



You’ll notice you can’t open in mongoDB compass locally



```
mongodb://localhost:27017/
```

```
connect ECONNREFUSED 127.0.0.1:27017
```

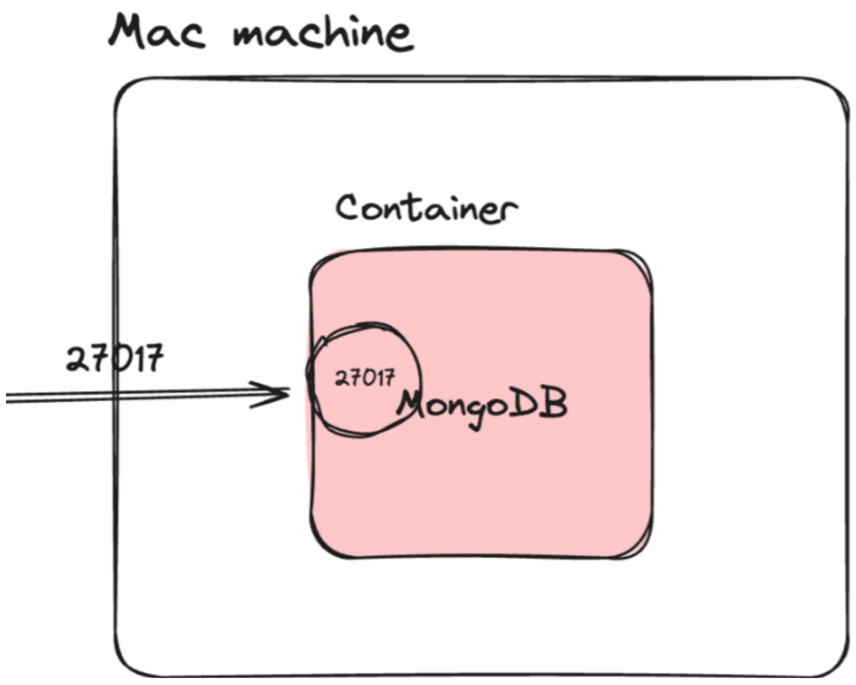
Why isn’t it unable to connect it , we havent told that when someone go for this port then it should access from this container

Currently mongoDB is running independently.

Adding a port mapping

The reason is that you haven’t added a port mapping

docker run -p 27017:27017 mongo



Starting in a detached mode

Adding -d will ensure it starts in the background

docker run -d -p 27017:27017 mongo

Now any request coming to machine 27017 port will go to the container's 27017 port we wont see any log

Inspecting a container

docker ps

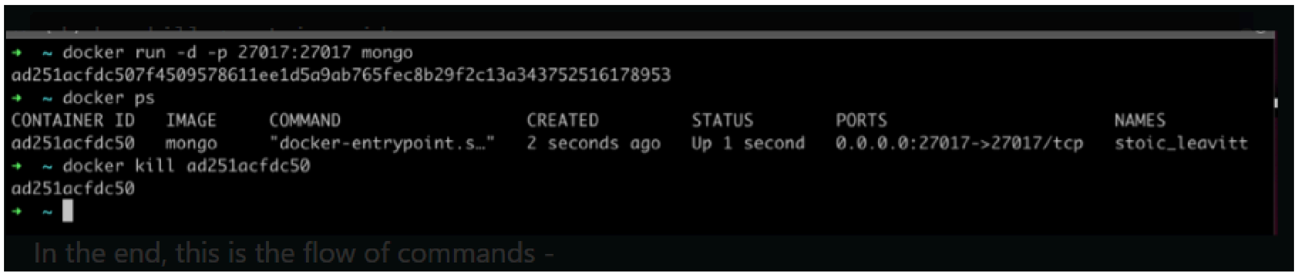
This will show all the container which are running currently

Stopping a container

docker kill <container_id>

Will stop the container that you are running

This is flow of commands—



```
➜  ~ docker run -d -p 27017:27017 mongo
ad251acfdc507f4509578611ee1d5a9ab765fec8b29f2c13a343752516178953
➜  ~ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                      NAMES
ad251acfdc50   mongo     "docker-entrypoint.s…"   2 seconds ago   Up 1 second    0.0.0.0:27017->27017/tcp   stoic_leavitt
➜  ~ docker kill ad251acfdc50
ad251acfdc50
➜  ~ 
```

In the end, this is the flow of commands -

Common packages

Postgres

docker run -e POSTGRES_PASSWORD=mysecretpassword -d -p 5432:5432 postgres

Connection string

postgresql://postgres:mysecretpassword@localhost:5432/postgres