Docker docker dockah!

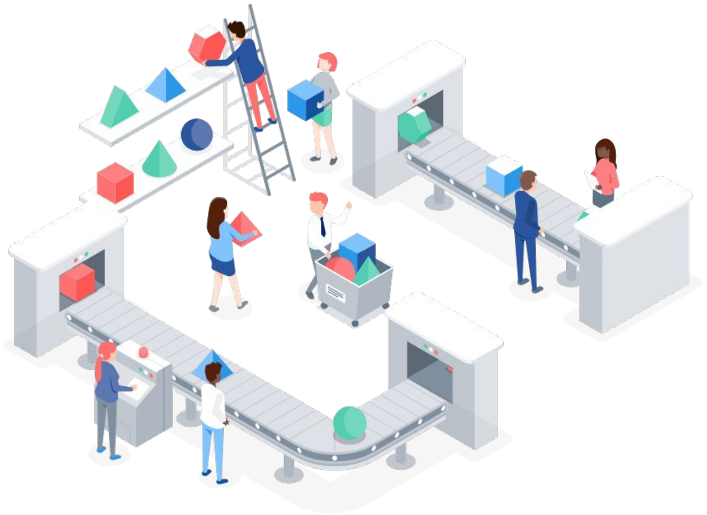
Dockah dockah dockah dockah

First of,

Install docker from docker website.
don’t worry, it supports Mac, Windows and Linux.

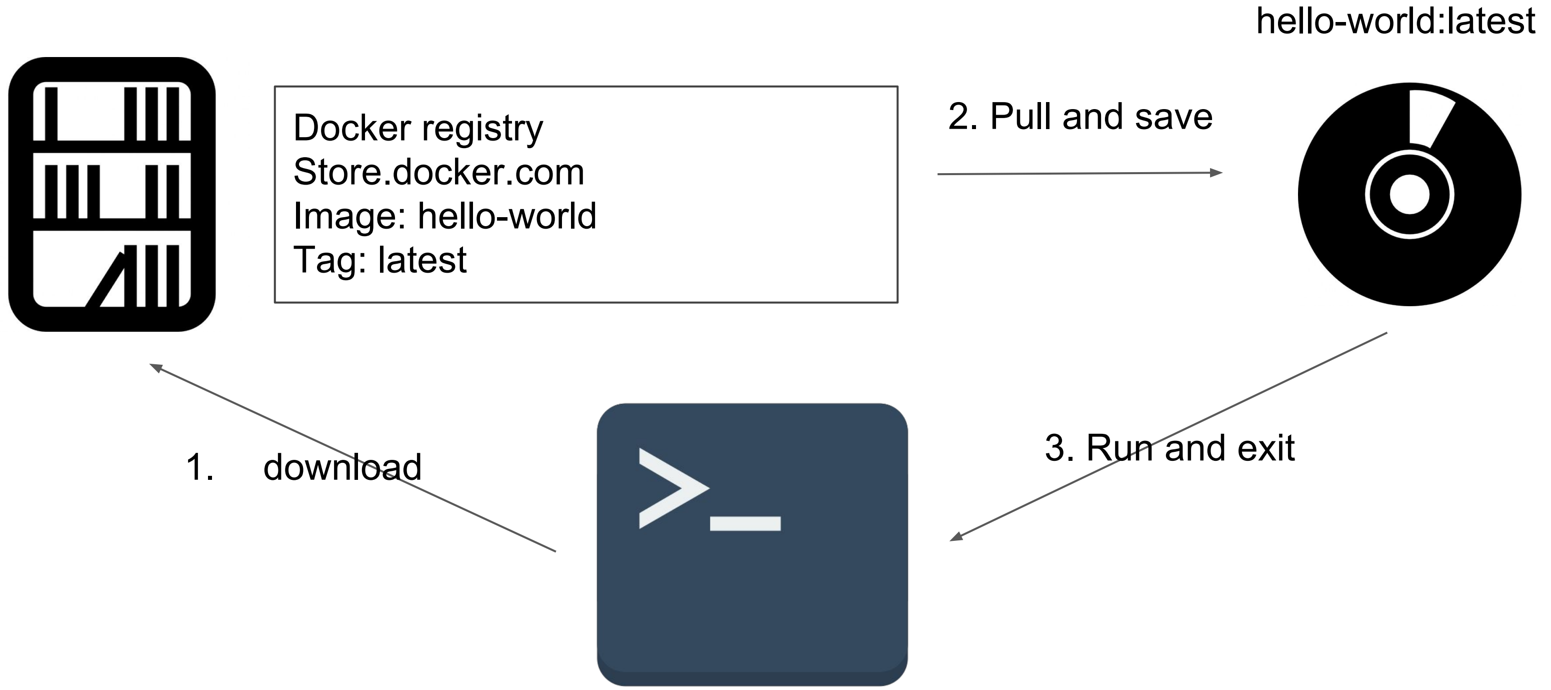
https://store.docker.com/search?type=edition&offering=community

Or simply google, ‘docker community’.



Running of first container

```
docker container run hello-world
```



Docker images

```
docker image pull alpine
```

```
docker image ls
```

```
REPOSITORY               TAG                  IMAGE ID             CREATED              VIRTUAL SIZE
alpine                  latest               c51f86c28340         4 weeks ago          1.109 MB
hello-world              latest               690ed74de00f         5 months ago         960 B
```

```
docker container run alpine ls -l
```

```
total 48
drwxr-xr-x    2 root     root          4096 Mar  2 16:20 bin
drwxr-xr-x    5 root     root           360 Mar 18 09:47 dev
drwxr-xr-x   13 root     root          4096 Mar 18 09:47 etc
drwxr-xr-x    2 root     root          4096 Mar  2 16:20 home
drwxr-xr-x    5 root     root          4096 Mar  2 16:20 lib
......
......
```

Docker run details

Docker engine

```
docker container run alpine ls -l
```

Alpine will lunch and run ls -l

My container

Alpine OS

Alpine container shutdown and output of ls -l will return to our OS

```
total 48
drwxr-xr-x    2 root     root
4096 Mar  2 16:20 bin
drwxr-xr-x    5 root     root
360 Mar 18 09:47 dev
......
......
```

Another example

```
docker container run alpine echo "hello from alpine"
```

```
hello from alpine
```

```
docker container run alpine /bin/sh
```

Another example

```
docker container run alpine echo "hello from alpine"
```

```
hello from alpine
```

```
docker container run alpine /bin/sh
```

How to get into bash inside alpine? Actually we already did, but it return ‘’

Another example

```
docker container run alpine echo "hello from
alpine"
```

```
hello from alpine
```

```
docker container run alpine /bin/sh
```

How to get into bash inside alpine? Actually we already did, but it return ‘’

```
docker container run -it alpine /bin/sh
```

to list containers

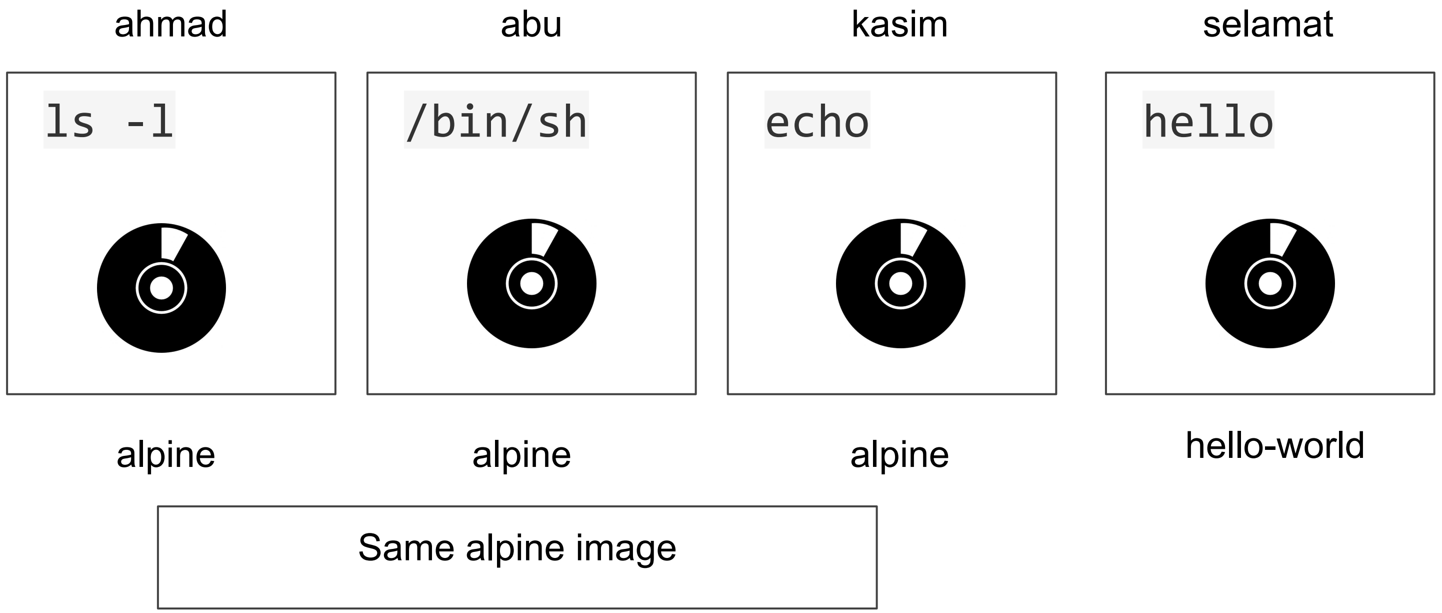
```
docker container ls
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
```

```
docker container ls -a
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                         PORTS
NAMES
36171a5da744        alpine              "/bin/sh"                5 minutes ago       Exited (0) 2 minutes ago
fervent_newton
a6a9d46d0b2f        alpine             "echo 'hello from alp"    6 minutes ago       Exited (0) 6 minutes ago
lonely_kilby
```

Docker container instances



Container isolation

```
docker container run -it alpine /bin/ash
```

```
echo "hello world" > hello.txt
```

```
 ls
```

```
docker container run alpine ls
```

Container isolation

```
docker container run -it alpine /bin/ash
```

```
echo "hello world" > hello.txt
```

```
 ls
```

```
docker container run alpine ls
```

Where is our hello.txt? missing!

Container isolation

```
docker container run -it alpine /bin/ash
```

```
echo "hello world" > hello.txt
```

```
 ls
```

```
docker container run alpine ls
```

Where is our hello.txt? missing!

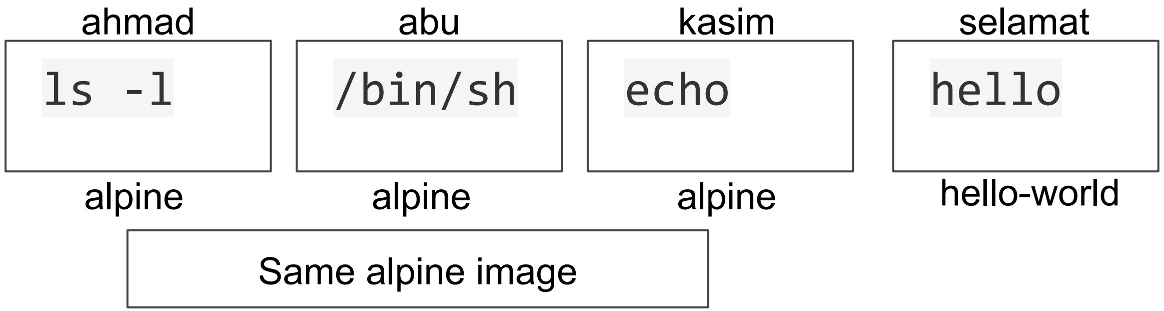
Actually, we run a new container using the OS image (alpine).

It is totally running a new OS!

Container isolation (cont)

```
docker container ls -a
```

```
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS                      PORTS
NAMES
36171a5da744        alpine              "ls"                2 minutes ago       Exited (0) 2 minutes ago
distracted_bhaskara
3030c9c91e12        alpine              "/bin/ash"          5 minutes ago       Exited (0) 2 minutes ago
fervent_newton
```



To start same container

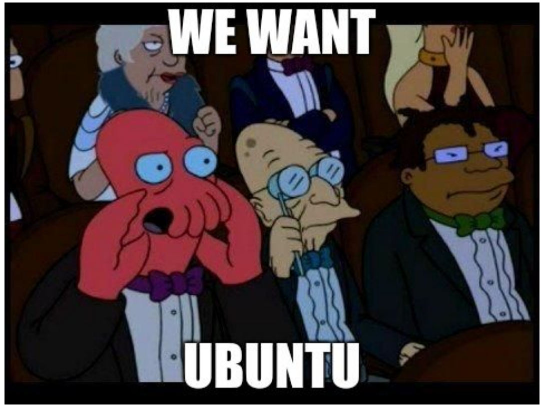
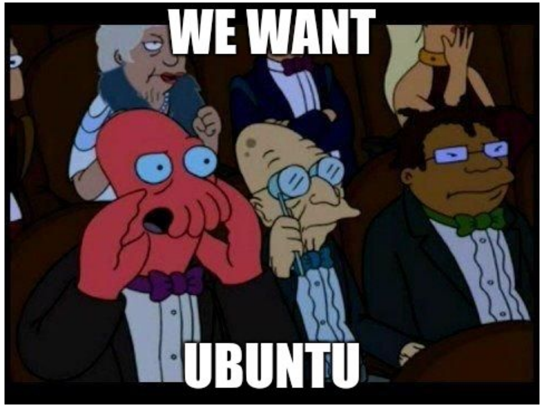
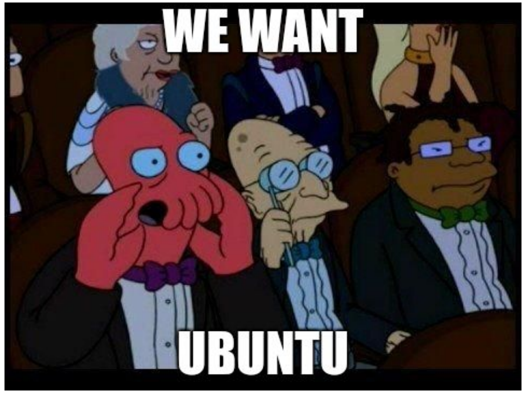
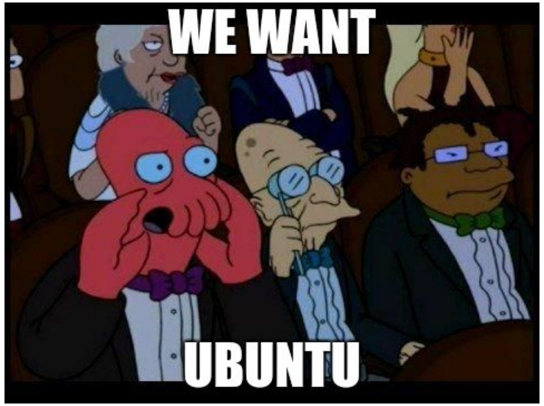
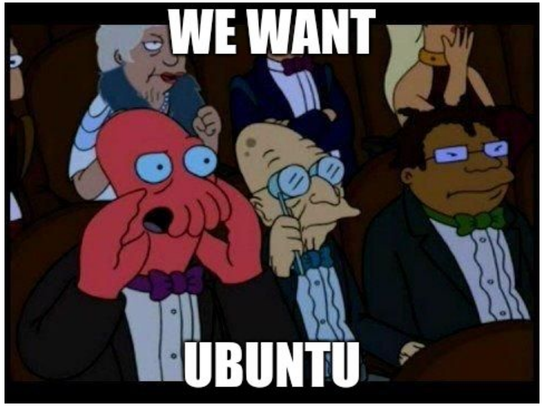
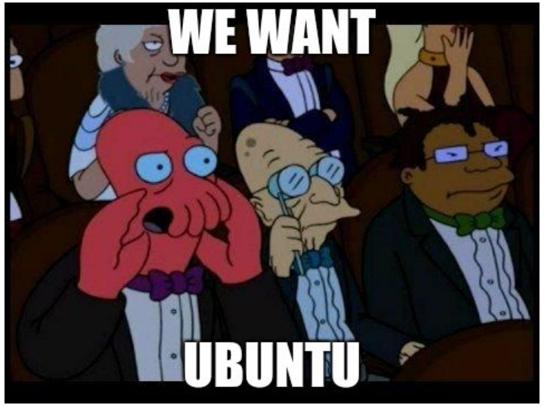
```
docker container start <container ID>
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                        PORTS
NAMES
3030c9c91e12        alpine              "/bin/ash"                2 minutes ago       Up 14 seconds
distracted_bhaskara
```

```
docker container exec <container ID> ls
```

Notes

- ***images*** - The file system and configuration of our application which are used to create containers. To find out more about a Docker image, run `docker image inspect alpine`. In the demo above, you used the `docker image pull` command to download the **alpine** image. When you executed the command `docker container run hello-world`, it also did a `docker image pull` behind the scenes to download the **hello-world** image.
- ***Containers*** - Running instances of Docker images — containers run the actual applications. A container includes an application and all of its dependencies. It shares the kernel with other containers, and runs as an isolated process in user space on the host OS. You created a container using `docker run` which you did using the alpine image that you downloaded. A list of running containers can be seen using the `docker container ls` command.
- ***Docker daemon*** - The background service running on the host that manages building, running and distributing Docker containers.
- ***Docker client*** - The command line tool that allows the user to interact with the Docker daemon.
- ***Docker Store*** - Store is, among other things, a registry of Docker images. You can think of the registry as a directory of all available Docker images. You'll be using this later in this tutorial.


WE WANT
UBUNTU

WE WANT
UBUNTU

WE WANT
UBUNTU

WE WANT
UBUNTU

WE WANT
UBUNTU

WE WANT
UBUNTU

Lets grab some ubuntu

```
docker container run -ti ubuntu bash
```

```
apt-get update
apt-get install -y figlet
figlet "hello docker"
```

```
exit
```

```
docker container ls -a
```

```
docker container commit CONTAINER_ID
```

Lets grab some ubuntu

```
docker image ls
```

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>              a104f9ae9c37        46 seconds ago
160MB
ubuntu              latest              14f60031763d        4 days ago
120MB
```

```
docker tag <id> ourfiglet
```

Image creation

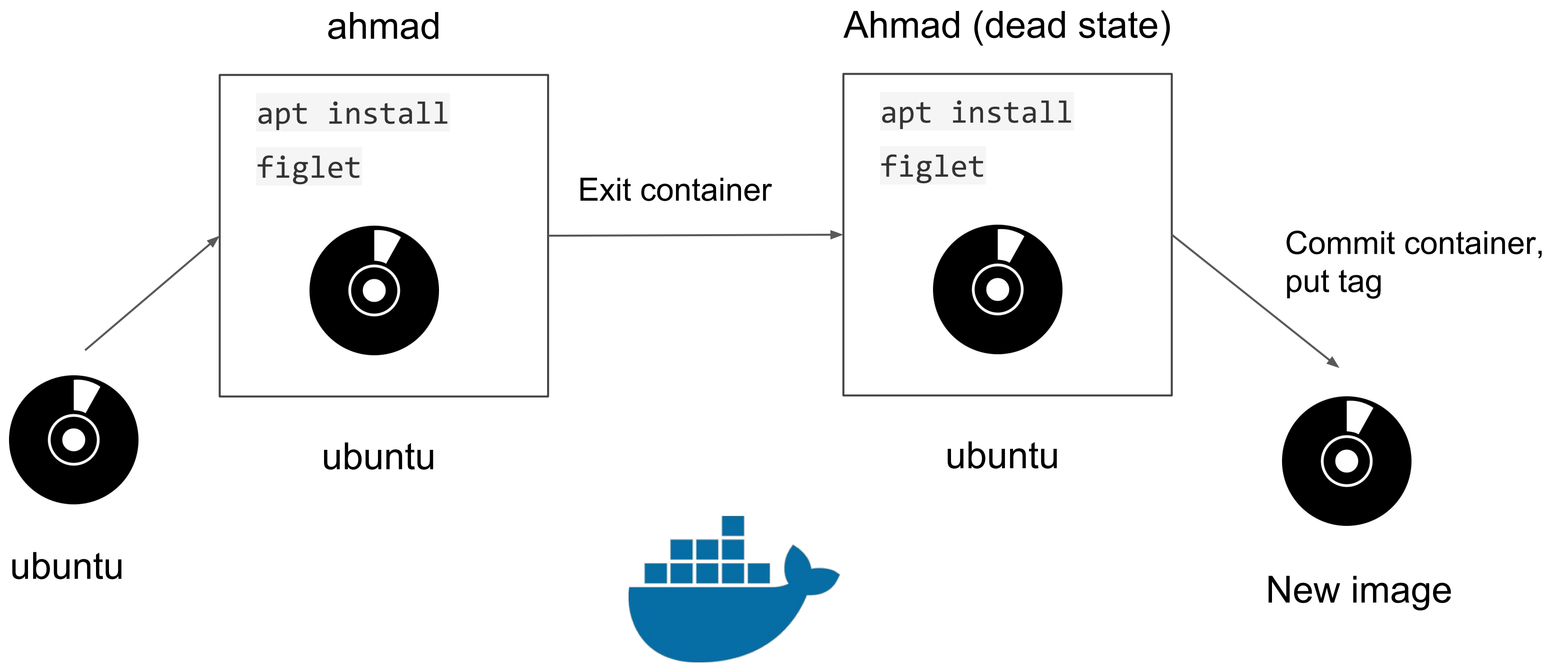


Image creation (cont)

```
docker container run ourfiglet figlet hello
```

```
 _          _ _
| |__   ___| | | ___
| '_ \ / _ \ | |/ _ \
| | | |  __/ | | (_) |
|_| |_|\___|_|_|\___/
```

Let’s create some simple Flask application

Create a new file called, app.y

```
from flask import Flask

app = Flask(__name__)

@app.route('/')

def hello_world():

    return 'Hey, we have Flask in a Docker container!'

if __name__ == '__main__':

    app.run(debug=True, host='0.0.0.0',port=5000)
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base
```

```
RUN apt-get update
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base

RUN apt-get update

RUN apt-get install -y python3 python3-pip
python3-wheel
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base

RUN apt-get update

RUN apt-get install -y python3 python3-pip
python3-wheel

RUN pip3 install Flask
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base

RUN apt-get update

RUN apt-get install -y python3 python3-pip
python3-wheel

RUN pip3 install Flask

EXPOSE 5000:5000
```

Let’s we create our own Flask image

Create a new file called, Dockerfile

```
FROM ubuntu:16.04 AS base

RUN apt-get update

RUN apt-get install -y python3 python3-pip python3-wheel

RUN pip3 install Flask

EXPOSE 5000:5000

WORKDIR /app

COPY . /app
```

And to run the Flask app

In the same file, Dockerfile

```
CMD [“python3”, “app.py”]
```

And to run the Flask app

In the same file, Dockerfile

```
CMD [“python3”, “app.py”]
```

```
docker image build -t flask:v0.1 .
```

```
docker container run -p 5000:5000 flask:v0.1
```

Now open a browser, go to localhost:5000