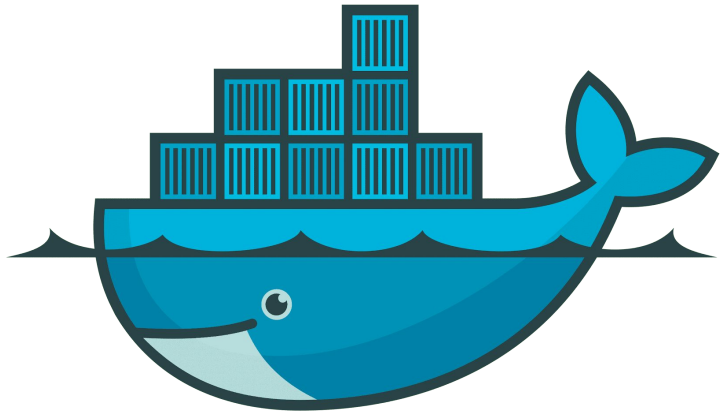
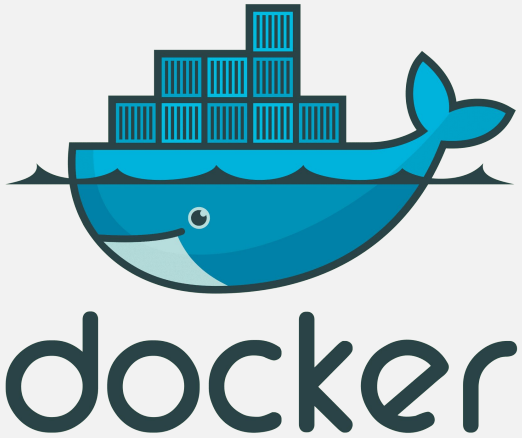


docker

Workshop

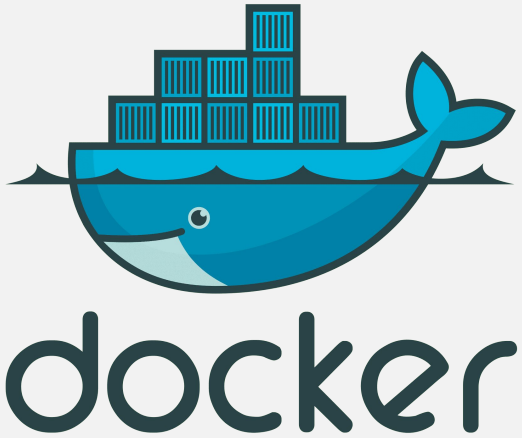


Agenda

1. What is Docker?
2. What is a Container?
3. Why use Docker?
4. Getting Started with Docker
5. Managing Containers & Images
6. Docker Images & Docker Hub
7. Docker Compose
8. Exercise 1
9. Further Learning

1

What is Docker?

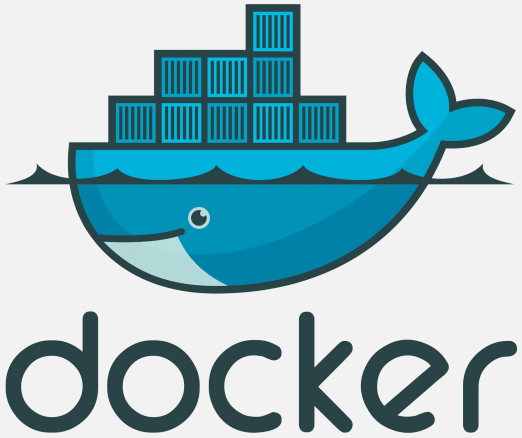


What is Docker?

- An open platform
- Made for developing, shipping, and running applications
- Separates applications from infrastructure
- Uses containers to achieve seperation

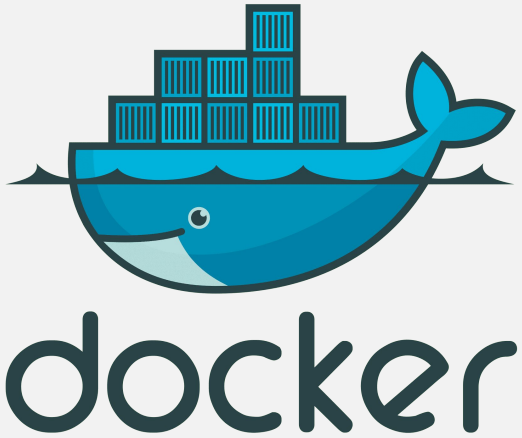
2

What is a Container?



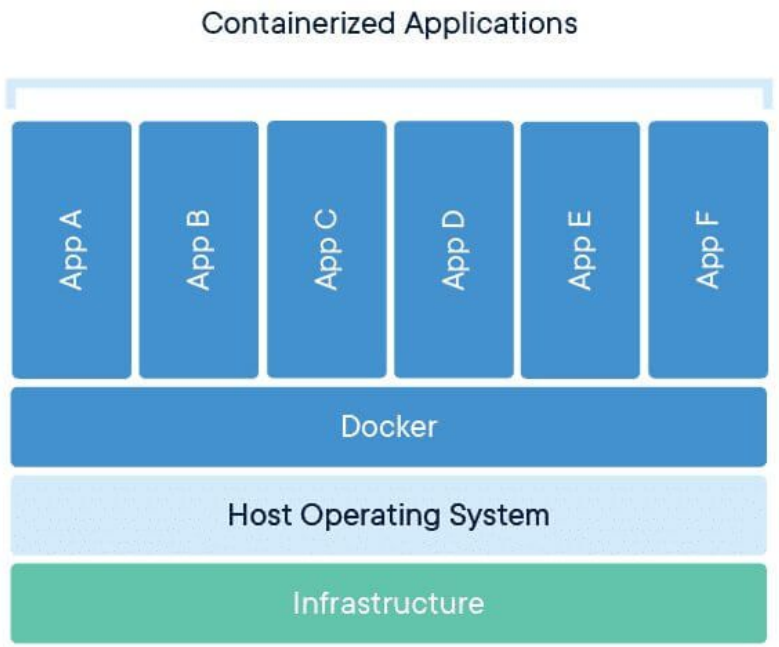
What is a Container?

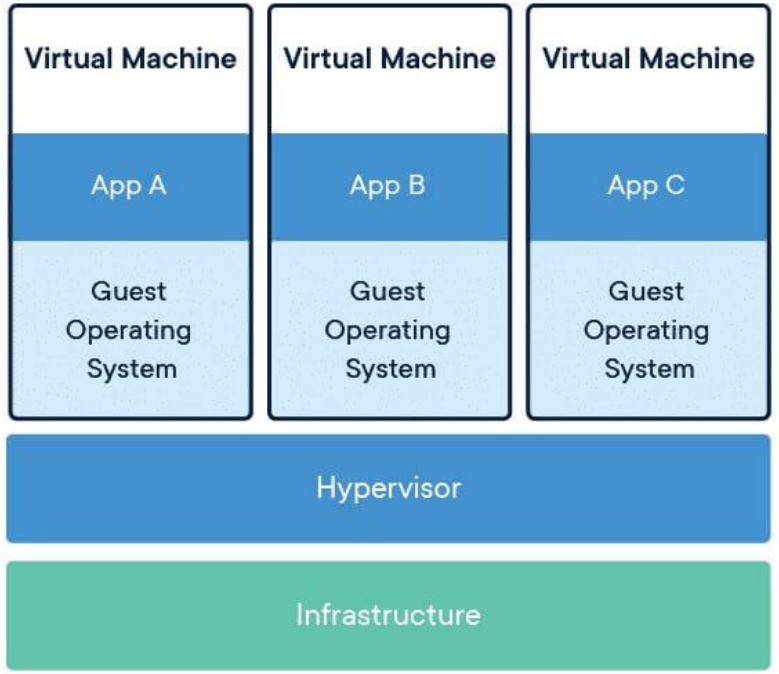
- Think of it as a Virtual Machine
- Lighter than a VM
- Contains the app and its dependencies
- Uses less resources than VMs
- Separates apps from host OS



What is a Container?

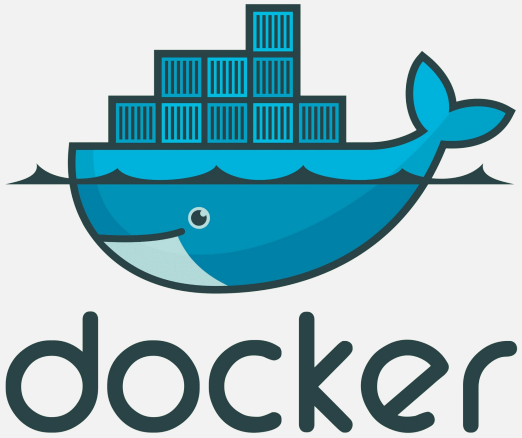
- Container VS VM





3

Why use Docker?



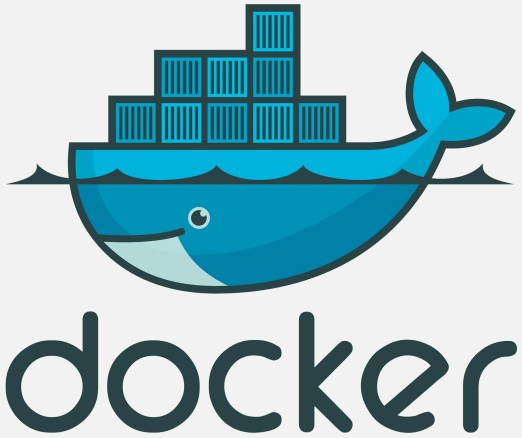
Why use Docker?

- Solves the infamous:

  “But it works on my machine!”
- Ensures apps work the same everywhere
- Simplifies deployment & scaling
- Reduces conflicts
- Saves resources

4

Getting Started with Docker

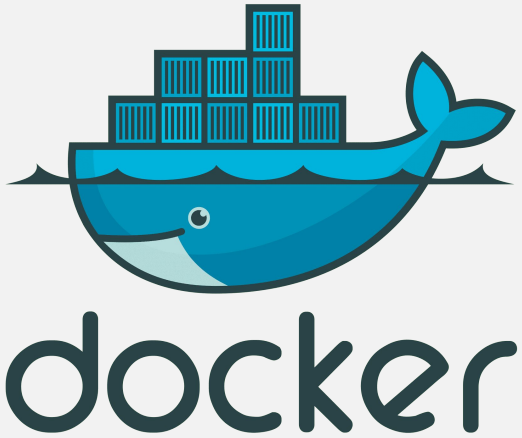


Getting Started with Docker

- Run our first ever container:

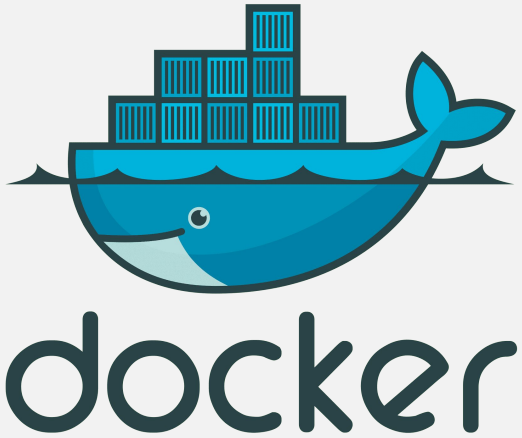
```
docker run -d -p 8080:80
docker/getting-started
```

- Open your browser
- Navigate to `localhost:8080`
- Congratulations! You just ran your first container



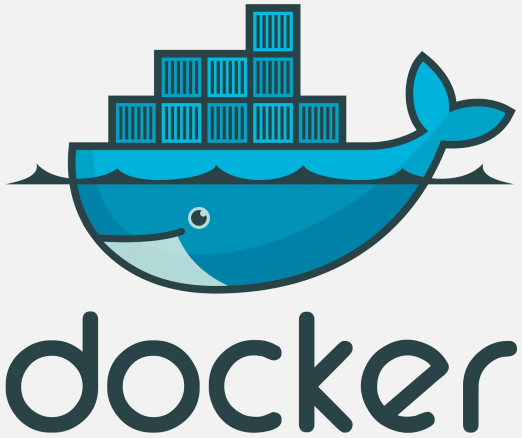
Getting Started with Docker

- Now, lets go over the command
- `docker run`
  - Tells the docker daemon to run something
- `-d`
  - Tells the docker daemon to run the container detached



Getting Started with Docker

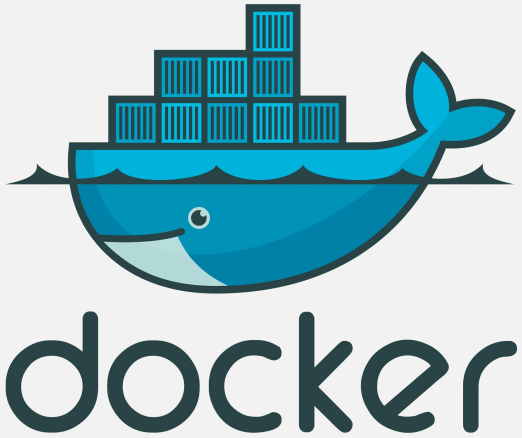
- `-p`
  - Tells the docker daemon how to port forward the container
- `8080:80`
  - Map the port 80 in the container to the port 8080 on the host



Getting Started with Docker

- `docker/getting-started`
  - The image we want to run

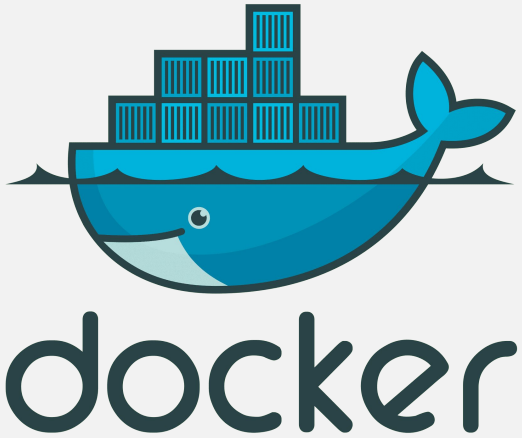
Managing Containers & Images



Managing Containers & Images

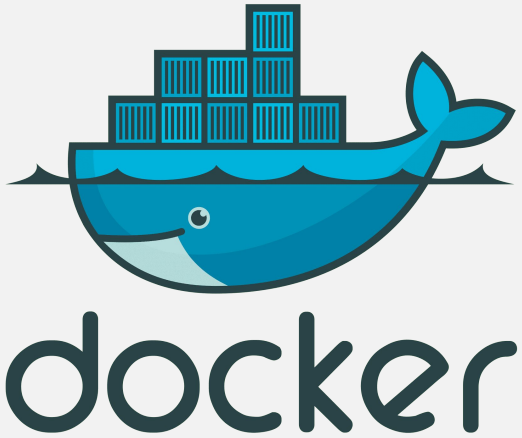
- **List containers**
  - `docker ps -a`
- **Stop container**
  - `docker stop <container_id>`
- **Remove container**
  - `docker rm <container_id>`
- **Remove image**
  - `docker rmi <image_name>`
- **View logs**
  - `docker logs <container_id>`

Docker Images & Docker Hub



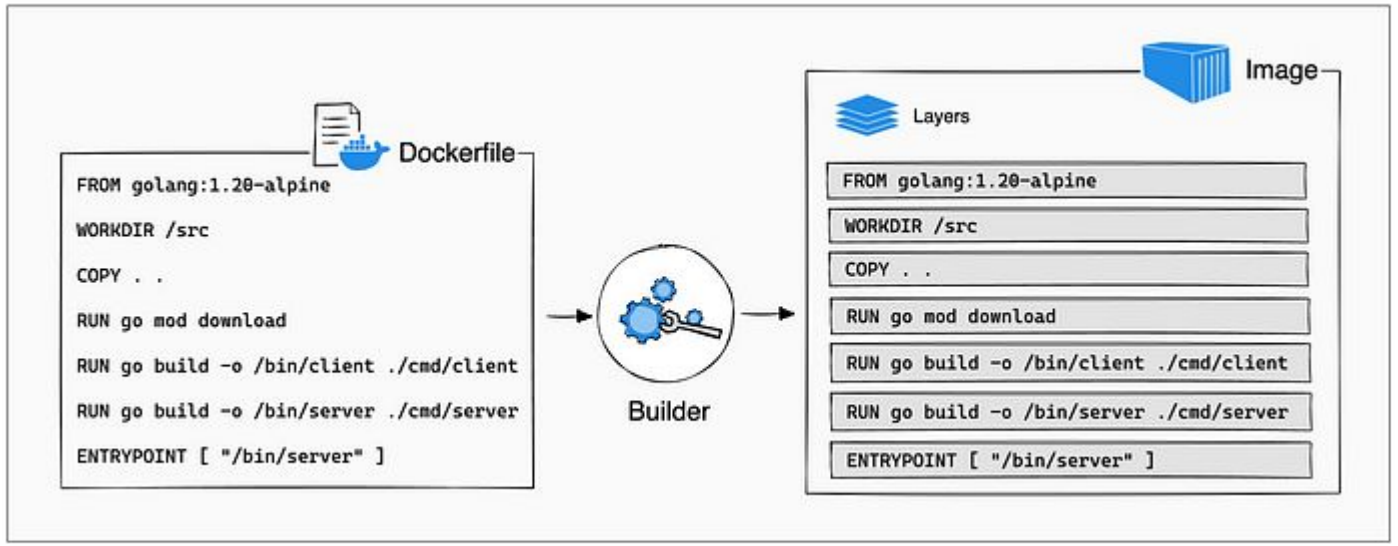
Docker Images & Docker Hub

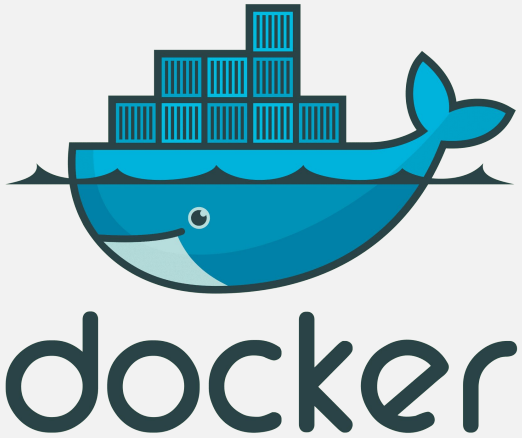
- Docker Image: A snapshot of an application and its environment.
- Docker Hub: Public registry for storing images.
- Build your own image:
  - `docker build -t <image_name> .`
- Pull existing images from Docker Hub:
  - `docker pull <image_name>`



Docker Images & Docker Hub

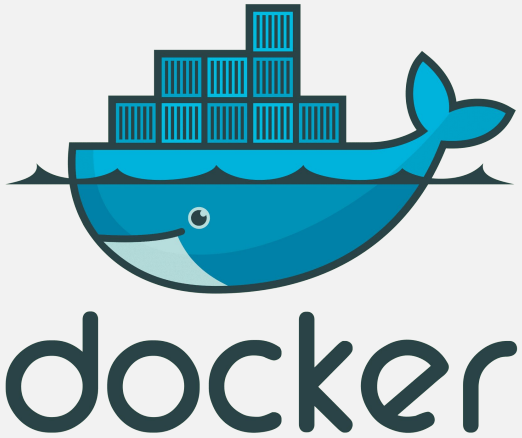
- Docker Image Layers





Docker Images & Docker Hub

- We will build a new image manually
- This process uses a new file called Dockerfile
- A Dockerfile is a set of instructions for the image builder
- A link will be sent in the WhatsApp group

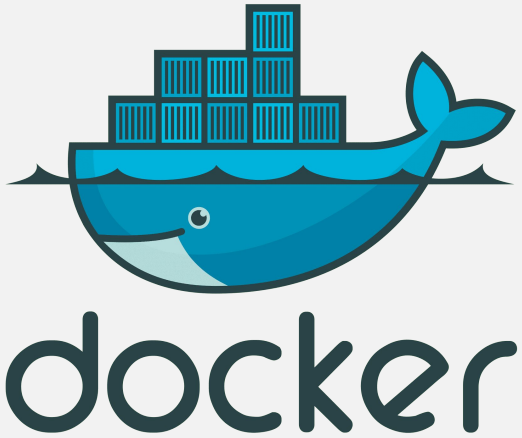


Docker Images & Docker Hub

- After downloading the file, extract the zip
- When extraction is done, navigate to the folder with a file called `package.json`
- Create a new file called `Dockerfile`
- Add this line to the top of the file:

```
# syntax=docker/dockerfile:1
```

- The line defines which dockerfile version to use



Docker Images & Docker Hub

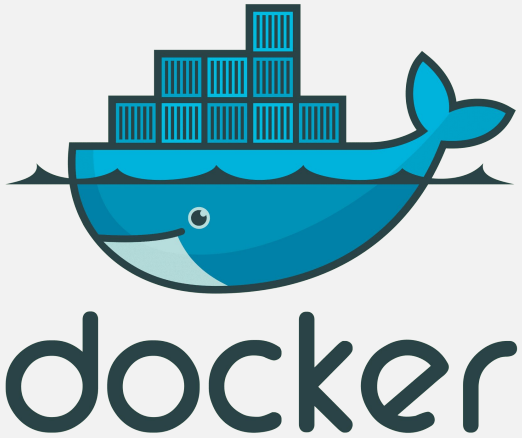
- On a new line, enter the following line:

```
FROM node:18-alpine
```

- This specifies the base image
- Next line, enter:

```
WORKDIR /app
```

- This line tells the image the working directory of the container



Docker Images & Docker Hub

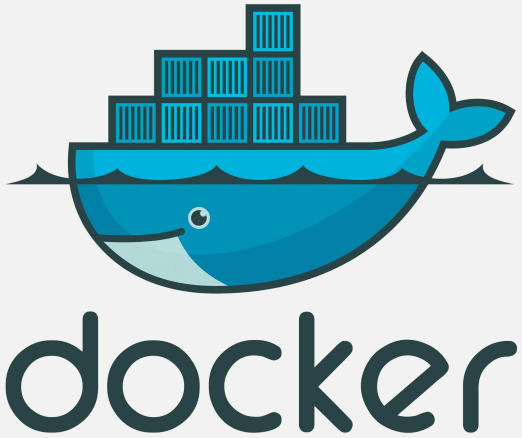
- Next line, enter:

```
COPY . .
```

- This copies all files from the host at the current directory to container
- Next line, enter:

```
RUN yarn install --production
```

- This line tells the container to run the command `yarn install`

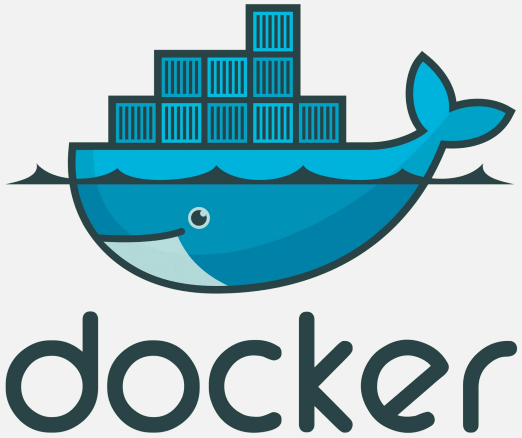


Docker Images & Docker Hub

- Finally, enter:

```
CMD [“node”,”./src/index.js”]
```

- This line specifies the command to run every time the container starts
- Take note that the application runs on port 3000



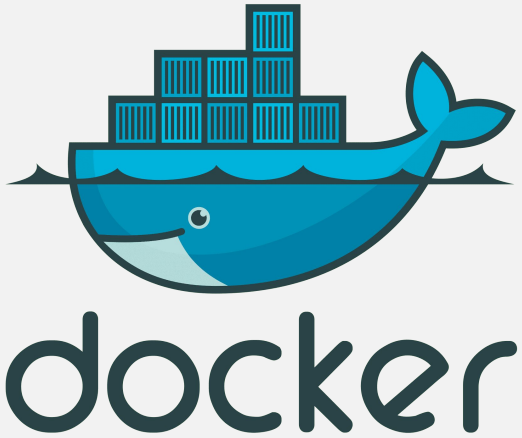
Docker Images & Docker Hub

- Now that the Dockerfile is done, we need to build the image
- To build the image, run:

```
docker build . -t my-first-image
```

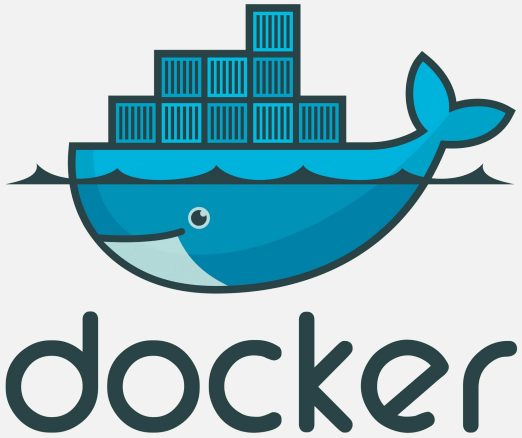
- What this command does is build the image
- Uses the `-t` flag to name the image
- The name is `my-first-image`

Docker Compose



Docker Compose

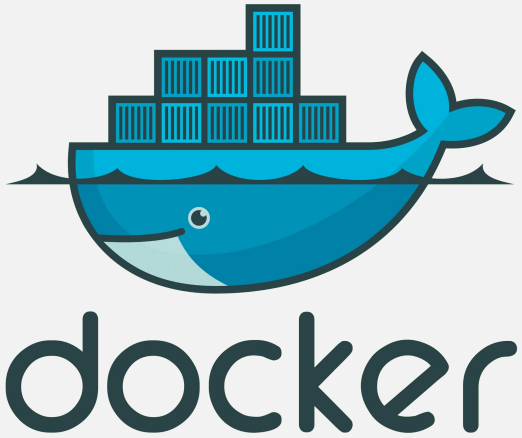
- Tool to define multi-container apps
- Use a `compose.yaml` file to specify services
- Start it using a single command:
  - `docker compose up -d`
- `-d` flag for detached



Docker Compose

- A link will be sent in the group chat for the docker compose file
- Download the file
- Move the file to same directory where we wrote Dockerfile
- Deploy it using `docker compose up -d`

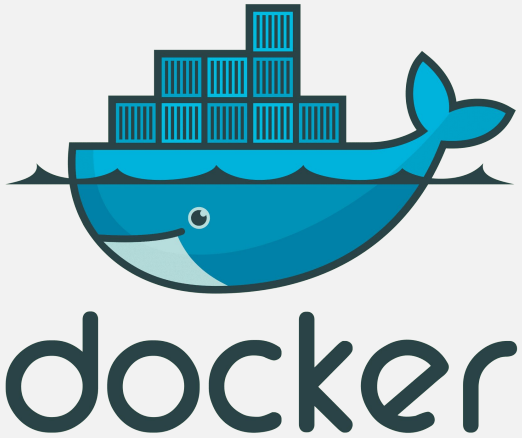
Exercise 1



Exercise 1

- Deploy the container created earlier
- Check the container logs
- Stop the container
- Start the container
- Stop & Remove the container
- Remove the image we built earlier

Further Learning



Further Learning

- https://docs.docker.com/get-started/workshop/ - Official Docker Docs
- https://www.youtube.com/watch?v=rIrNIzy6U_g - Fireship Channel