Toronto ML Microservices and API meetup

06.17.2020

Ike Okonkwo

Docker 101 - Part I


docker

Docker 101

Part I

- Introduction to Docker
- Common Docker Commands
- Docker + Microservices
- Dockerizing a HTTP service
- Questions / Demo

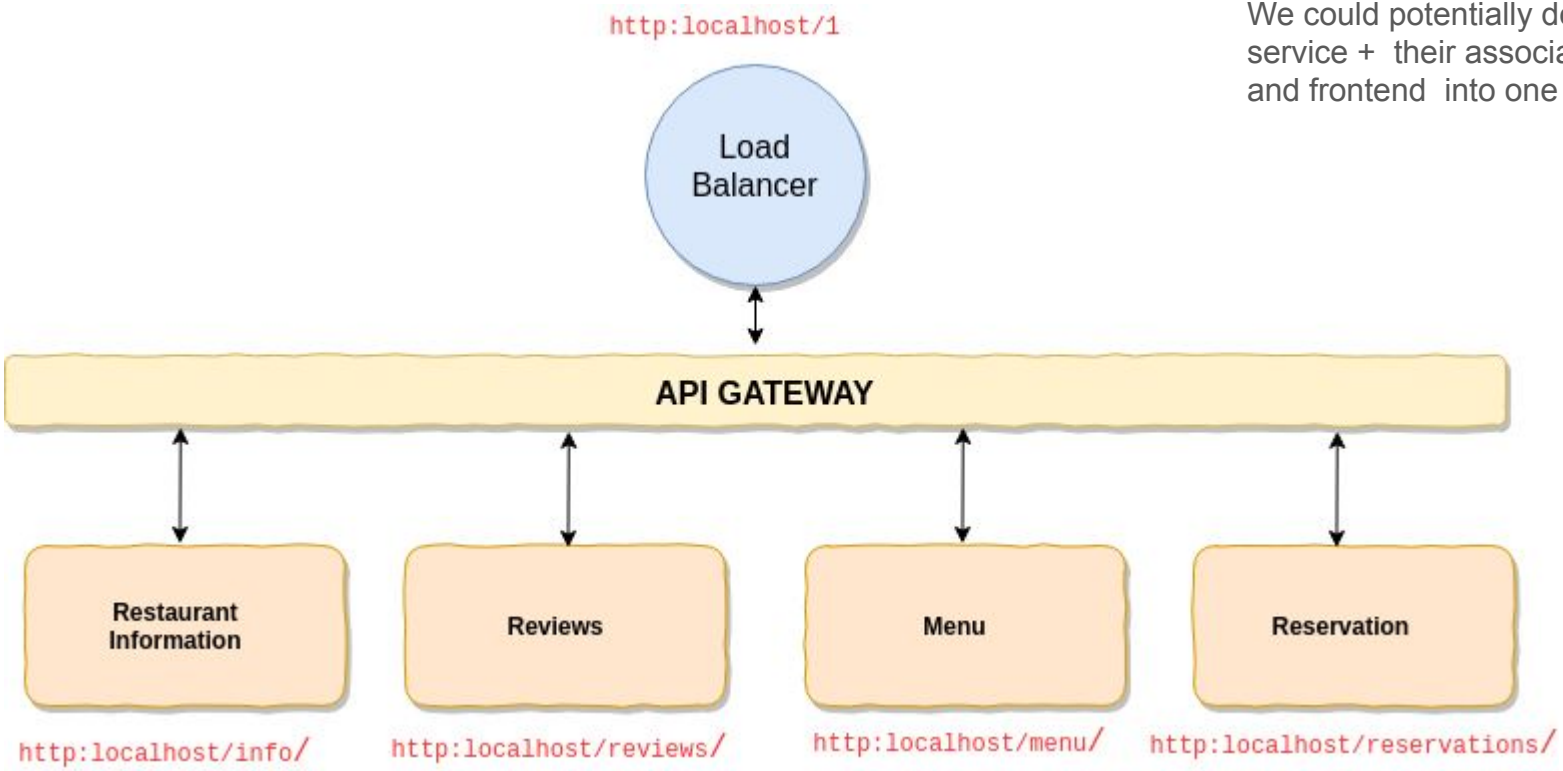
Part II

- Working with multiple containers
- Adding database support
- Deploying to production (AWS)
- Questions / Demo

Docker 101

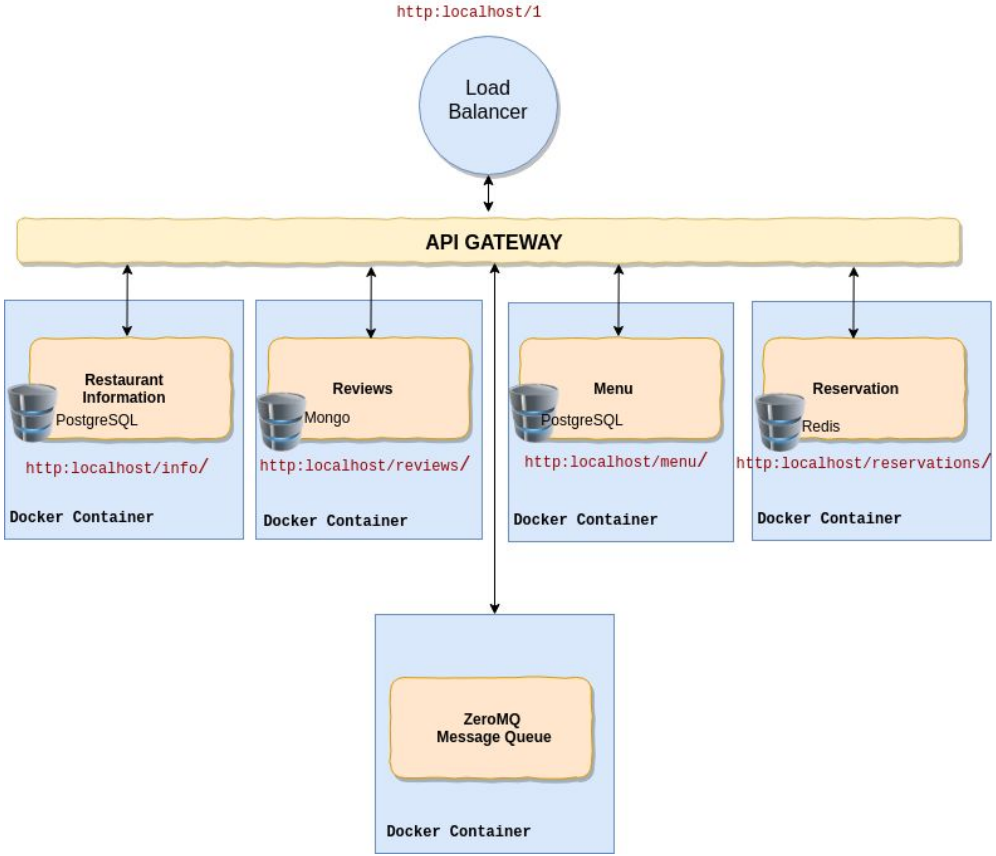
- Docker container are analogous to virtual environments
- They’re reproducible and can be deployed as part of a pipeline
- Each container is unique and isolated
- `Container` inherits from `Image`
- `Image` inherits from `DockerHub(registry)`

Docker + Microservices



We could potentially dockerize each service + their associated databases and frontend into one docker container

Docker + Microservices



Each service could have persistent storage appropriate for the data generated within the container

Common Docker commands - I

```
docker --version

docker version

docker info

docker images

docker ps -a

docker pull node:latest
```

Common Docker commands - II

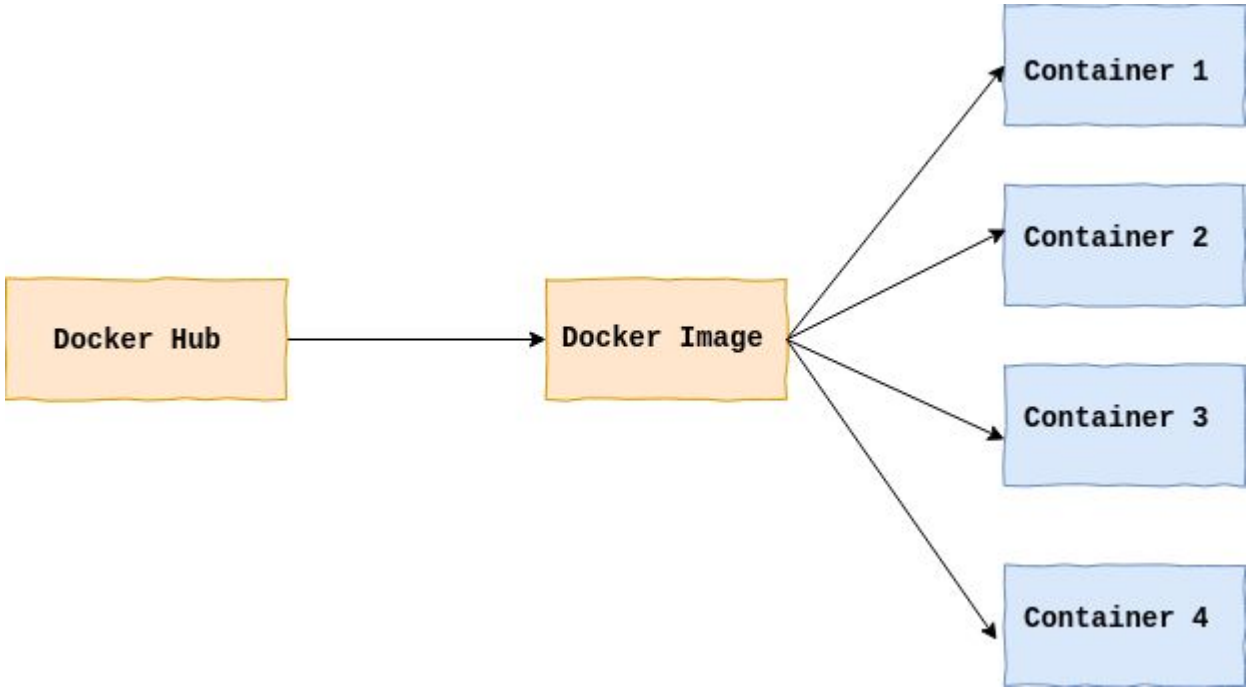
`docker run` - creates new container based on local image

`docker build` - create a new image based on a Dockerfile . Dockerfile is a config file for docker builds

`docker pull` - pulls image from docker Hub

```
docker pull <image name>:<version>
docker pull node:latest
```

Docker Registry / Images



Common Docker commands - III

There are a few other commands we should be familiar with : stop , start and restart

```
docker stop <container-name>
docker start <container-name>
```

As we create and use images and containers, these commands help remove them

```
docker rm <container-name>
docker rm <container-name>  <container-name> … <>
docker rm -f <container-name>
docker rmi <image-name>
```

Common Docker commands - IV

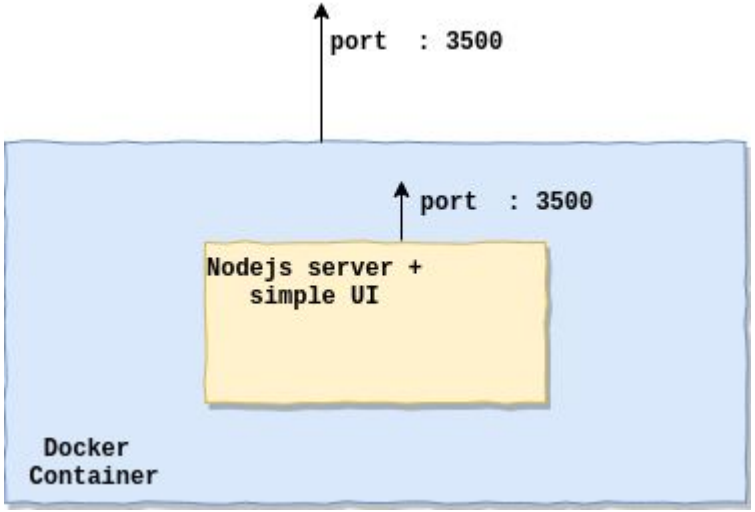
```
#delete all containers
docker rm $(docker ps -a -q)

# delete all images
docker rmi $(docker images -q)

# build image for microservice
docker build -t node_demo_service .

# run microservice
docker run -d -p 3500:3500 --rm node_demo_service

docker history node
```



Anatomy of Docker commands

```
docker build -t node_service .
         build : builds image
         -t : tag identifier


docker run -d -p 4000:4000 --rm node_service
         -d : run in detached mode
         -p : port number assignment < host : container >
         -rm : short lived container, deletes itself when the container is stopped
```

Anatomy of a Dockerfile

```
# What image do you want to start building on?
FROM node:latest

# Make a folder in your image where your app's source code can live
RUN mkdir -p /src/app

# Tell your container where your app's source code will live
WORKDIR /src/app

# What source code do you what to copy, and where to put it?
COPY . /src/app

# Does your app have any dependencies that should be installed?
RUN yarn install

# Expose port and start app
EXPOSE 4000
CMD [ "node", "main.js" ]
```

References

[1] https://github.com/dylanlrrb/Please-Contain-Yourself

[2] https://hub.docker.com/

Questions