

# Toronto ML Microservices and API meetup

07.29.2020

Ike Okonkwo

Docker - Part 2



docker

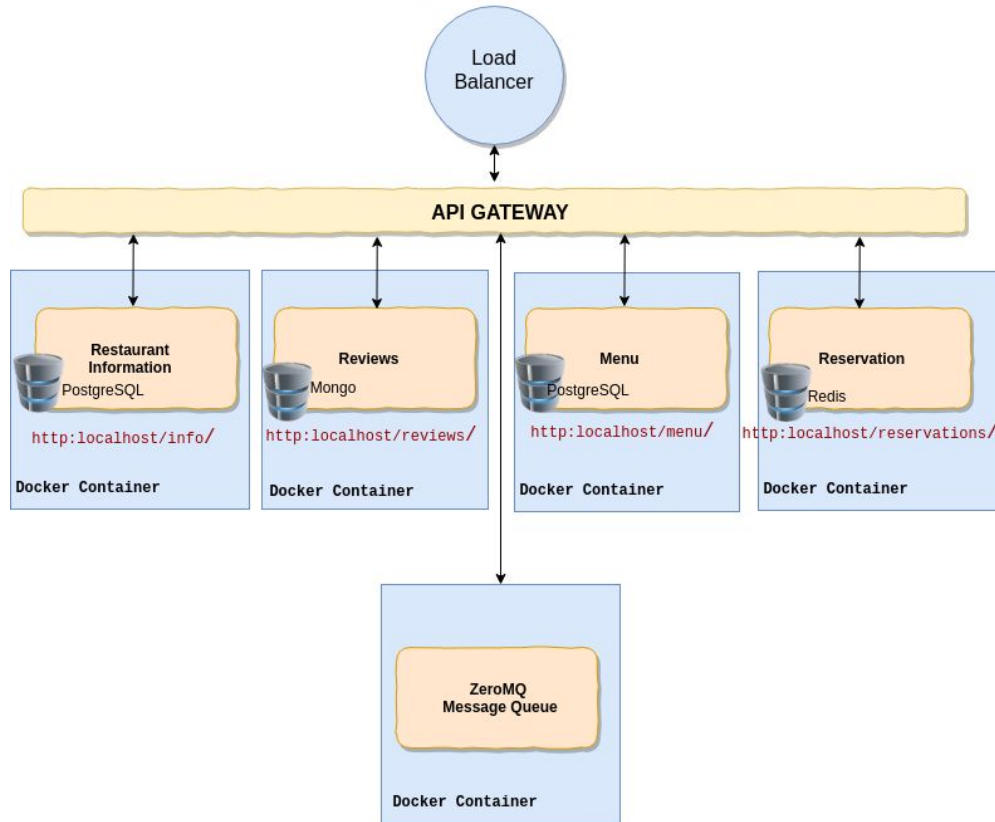
# Docker 102

## Part II

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

# Docker + Microservices

http://localhost/1



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

# Persist volume / data in a container

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

```
docker run <absolute path> : /home/ike/work  jupyter/scipy-notebook
```

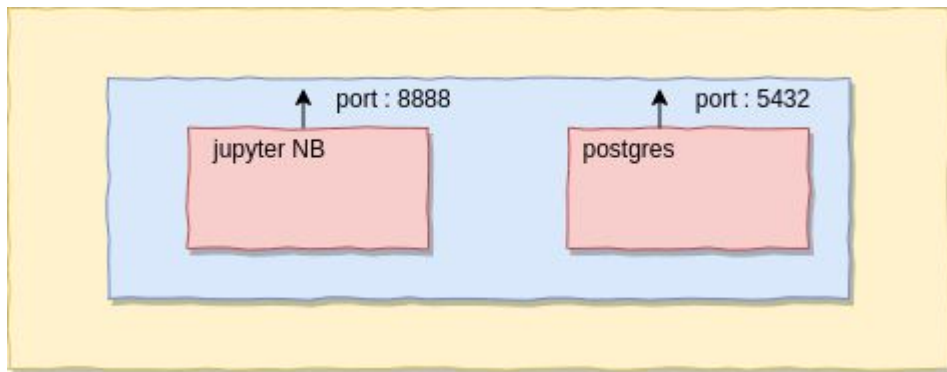
```
docker run -d -p 8888:8888 -v `pwd`: /home/ike/work --name demo jupyter/scipy-notebook
```

There are a few other commands we should be familiar with to create and mount volumes

```
docker create volume vol
```

```
docker run -v vol: /home/ike/work  jupyter/scipy-notebook
```

# Working with multiple containers



We can spin up each container separately

```
docker run -d -p 8888:8888 jupyter/scipy-notebook
```

```
docker logs <container ID>
```

Spin up the postgres container

```
docker run -d -p 5432:5432 postgres
```

```
docker volume create pgdata
```

```
docker run -d -p 5432:5432 -v pgdata:/var/lib/postgresql/data postgres
```

Examples used from <https://github.com/iaoeirner/beyond-jupyter>

# Docker Compose

Docker	Docker Compose
Single Container	Multiple Containers
Dockerfile	docker-compose.yml
docker run	docker-compose up

# Anatomy of a Docker Compose file

```
version: "3"
services:
  jupyter:
    image: jupyter/scipy-notebook
    container_name: "my_jupyter"
    ports:
      - 8888:8888
  postgres:
    image: postgres
    container_name: "my_postgres"
    ports:
      - 5432:5432
    environment:
      - SHARED_PASSWORD=password
      - POSTGRES_PASSWORD=secret
    volumes:
      - ./scripts/./docker-entrypoint-initdb.d/
      - pg_data:/var/lib/postgresql/data
volumes:
  pg_data:
```



# Deploying to AWS

```
#run docker compose
```

```
docker-compose up -d
```

```
# check logs
```

```
docker logs <container ID>
```

```
# shutdown services and remove volumes
```

```
docker-compose down -v
```

```
# inspect network
```

```
docker network ls
```

```
docker network inspect bridge
```

```
# push to registry / AWS deploy
```

```
docker push <username> / <docker-image>
```

# References

- [1] Examples used from <https://github.com/jgoerner/beyond-jupyter>
- [2] <https://hub.docker.com/>
- [3] <https://github.com/dylanlrrb/Please-Contain-Yourself>

# Questions

