Docker Compose

Objectives

We are going to learn ,

- How to launch Multi Container Apps
- How to create spec file for Compose
- How to scale Containers
- Docker Compose Operations

Fig => Compose

Fig was the original tool to launch multi container environments which is now been officially adopted by Docker Inc. and replaced with Docker Compose.

Compose

- Compose allows one to define a specification for multi container application using YAML and launch it as a stack.
- This file can then be version controlled and shared with others.

Use Cases

- Development Environments
- Automated Testing
- Production Deployments

Steps to use Compose

- Install Docker Compose (docker toolbox)
- Define docker-compose.yml file with multi container app specifications
- Run “docker-compose up” to bring up the entire app in one go

Group Exercise

We will create a compose file for wordpress app that we launched and linked manually earlier.

Compose File

- Create a working directory

  $ mkdir stack

  $ cd stack

- Create docker-compose.yml file with specifications as per the following link

https://gist.github.com/initcron/54fd41f4578de7babf48

Compose File Reference

To learn more about creating compose file, refer to the following resource

http://docs.docker.com/compose/compose-file/

Launch Stack

- Bring up the stack using docker-compose

```
$ docker-compose up &
```

Compose Operations

- List running containers

```
$ docker-compose ps
```

- Examine the logs

```
$ docker-compose logs
```

Scaling Containers

- Scale up wordpress containers

```
$ docker-compose scale wordpress=2
```

- Scale Down

```
$ docker-compose scale wordpress=1
```

Stop

- Stop container stack

```
$ docker-compose stop
```

- Remote Container Stack

```
$ docker-compose rm
```

Exercise

Create and launch docker compose stack to run two tier django application which consists of a web server with python app and postgresql database backend

https://docs.docker.com/compose/django/