Docker Compose

Agenda

What is docker compose?

Features

Installation

YML File

Commands

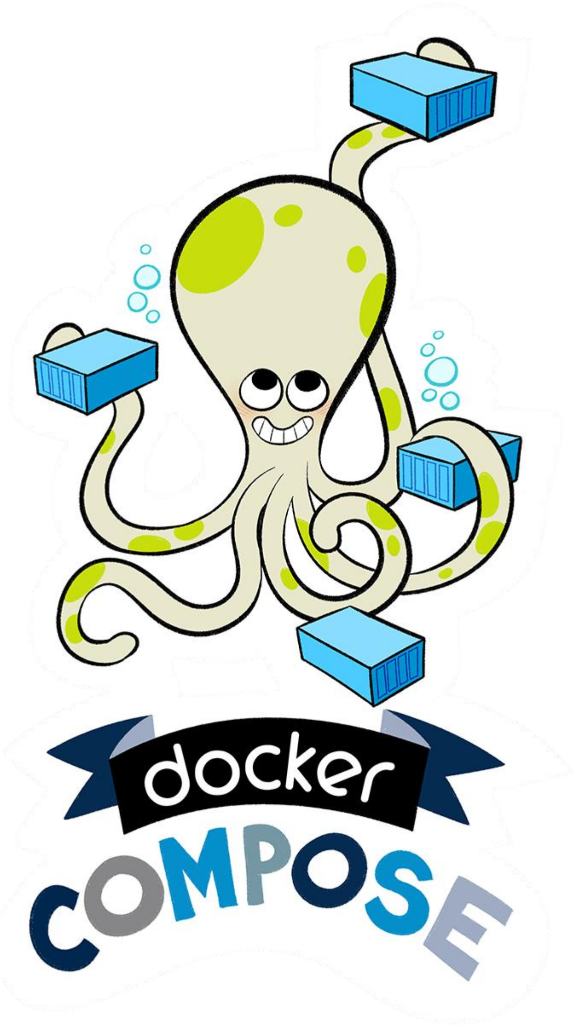
Lab 10: Simple Docker Compose

Dive into the YML File

Lab 11: Docker Compose

WHAT IS DOCKER COMPOSE?

- Docker Compose is a tool for defining and running complex applications with Docker.
- Define a multi-container application in a single file
- Spin your application up in a single command



FEATURES

- Multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Variables and moving a composition between environments
- Multiple compose files

DOCKER COMPOSE CONFIGURATION

- YAML is used to create docker compose configuration file.
- YAML: Yet Another Markup Language
- Whitespace indentation is used for denoting structure.
- Tab characters are *not allowed* as part of that indentation.
- Comments begin with the number sign (**#**), can start anywhere on a line and continue until the end of the line.
- List members are denoted by a leading hyphen (**-**) with one member per line.
- An associative array entry is represented using colon space in the form **key: value** with one entry per line.
- Strings are ordinarily unquoted but may be enclosed in double-quotes (**"**), or single-quotes (**‘**).

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
     template:
        metadata:
          labels:
            app: nginx
spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

DOCKER COMPOSE YML FILE FOR SQL SERVER

```yaml
services:
  mssql:
    image: mcr.microsoft.com/mssql/server:2022-latest
    container_name: sqlserver2k22
    ports:
    - "1433:1433"
    environment:
      ACCEPT_EULA: "Y"
      MSSQL_SA_PASSWORD: "YourStrongPassword123"
      MSSQL_DATA_DIR: "/var/opt/mssql/data"
      MSSQL_LOG_DIR: "/var/opt/mssql/log"
      hostname: sqlserver2k22

    restart: unless-stopped
    volumes:
    - mssql_data:/var/opt/mssql

volumes:
  mssql_data:
```

DOCKER COMPOSE YML FILE FOR SQL SERVER: ADDITIONAL PROPERTIES

```
environment:
      ACCEPT_EULA: "Y"
      MSSQL_SA_PASSWORD: "YourStrongPassword123"
      MSSQL_DATA_DIR: "/var/opt/mssql/data"
      MSSQL_LOG_DIR: "/var/opt/mssql/log“
      hostname: sqlserver2k22
      MSSQL_AGENT_ENABLED: "true"
      MSSQL_PID: "Developer"
      shm_size: '2gb'
      tz: "GMT+3"

deploy:
      resources:
        limits:
          cpus: ‘8.0'
          memory: 24G
        reservations:
          cpus: ‘4.0'
          memory: 16G
```

BEFORE DOCKER COMPOSE (*STARTING 4 CONTAINERS INDIVIDUALLY*)

```
$ docker run -d -it --name redis redis
$ docker run -d -it --name postgres linhmtran168/postgres
$ docker run -d -it --name web \ -v
~/Dev/gitlab.com/linhmtran168/test-project:/var/www/html \ --
link postgres:db --link redis:redis linhmtran168/php-web
$ docker run -d -it -p 80:80 --name nginx \ --link web:web --
volumes-from web linhmtran168/php-nginx
$ docker run -d -it --name node --link web:web \ --volumes-from
web linhmtran168/gulp-bower
```

AFTER DOCKER COMPOSE (*SINGLE FILE WITH ALL CONTAINERS*)

```
web:
  build: .
  links:
    - redis:redis
    - postgres:db
  volumes:
    - .:/var/www/html

  nginx:
  build: ../docker-php-nginx
  ports:
    - "80:80"
  links:
    - web:web
  volumes_from:
    - web
```

COMMANDS

- Create and start all the containers listed in the “docker-compose.yml”

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

- List all the containers belong to the compose environment instance:

```
$ docker-compose ps
```

SIMPLE DOCKER COMPOSE

Lab

https://gitlab.com/docker-workshop/Lab-11