



Module 11: Docker Compose

Docker Workshop



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

Installation

🚀 Windows / Mac - Docker Compose is included 😊

🚀 Linux:

```
sudo curl -L  
https://github.com/docker/compose/releases/download/1.21.2/docker-  
compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

Test the installation

```
docker-compose --version
```

Before Docker Compose

```
$ 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

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



YML File

- ✦ Version
- ✦ Services
 - ✦ Build
 - ✦ Image
 - ✦ Environment
 - ✦ Ports
 - ✦ Volumes
- ✦ Volumes
- ✦ Networks

Simple YAML File

version: '3'

services:

jenkins:

image: jenkins/jenkins:its

ports:

- "8080:8080"**
- "50000:50000"**

artifactory:

image: docker.bintray.io/jfrog/artifactory-oss:latest

ports:

- 8081:8081**

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
```

- ✦ Sets the number of containers:

```
$ docker-compose scale web=3
```

Commands

build	Build or rebuild services
help	Get help on a command
kill	Kill containers
logs	View output from containers
port	Print the public port for a port binding
ps	List containers
pull	Pulls service images
rm	Remove stopped containers
run	Run a one-off command
scale	Set number of containers for a service
start	Start services
stop	Stop services
restart	Restart services
up	Create and start containers

Docker Compose

Demo



Lab 10: Simple Docker Compose

Lab



<https://gitlab.com/docker-workshop/Lab-10>

Dive into the YAML File

✦ Image / Build

- ✦ image: jenkins/jenkins:its
- ✦ build: ./dir

```
version: '3'
```

```
services:
```

```
  jenkins:
```

```
    image: jenkins/jenkins:its
```

```
    ports:
```

```
      - "8080:8080"
```

```
      - "50000:50000"
```

```
  myapp:
```

```
    build: ./dir
```

Dive into the YAML File

✦ **Depends_on**

✦ web:
 build: .
 depends_on:
 - db
 - redis

```
version: '3'
```

```
services:
```

```
  web:
```

```
    build: .
```

```
    depends_on:
```

```
      - db
```

```
      - redis
```

Dive into the YAML File

✦ Restart

- ✦ restart: "no"
- ✦ restart: always
- ✦ restart: on-failure
- ✦ restart: unless-stopped

```
version: '3'
```

```
services:
```

```
  jenkins:
```

```
    image: jenkins/jenkins:its
```

```
    ports:
```

```
      - "8080:8080"
```

```
      - "50000:50000"
```

```
    restart: always
```


Dive into the YAML File

⚡ Environment

- ⚡ MYSQL_USER: wordpress
- ⚡ MYSQL_PASSWORD: wordpress

⚡ Environment File

- ⚡ env_file: .env

version: '3'

services:

jenkins:

image: jenkins/jenkins:its

ports:

- "8080:8080"

- "50000:50000"

restart: always

environment:

MYSQL_USER: wordpress

env_file: .env

Dive into the YAML File

⚡ Networks

⚡ web:
 networks:
 - sela

⚡ networks:
 sela:
 other:

```
version: '3'
```

```
services:
```

```
  jenkins:
```

```
    image: jenkins/jenkins:its
```

```
    ports:
```

```
      - "8080:8080"
```

```
      - "50000:50000"
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_USER: wordpress
```

```
    networks:
```

```
      - sela
```

```
networks:
```

```
  sela:
```

```
  other:
```

Dive into the YAML File

⚡ Volumes

⚡ volumes:
- "dbdata:/var/lib/postgresql/data"

⚡ volumes :
dbdata:

version: '3'

services:

jenkins:

image: jenkins/jenkins:its

ports:

- "8080:8080"

- "50000:50000"

restart: always

environment:

MYSQL_USER: wordpress

networks:

- sela

volumes:

- dbdata:/var/lib/postgresql/data

networks:

sela:

other:

Volumes:

dbdata:

Docker Compose

Demo



Lab 11: Docker Compose

Lab



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

Questions

