EvokeTechnologies

DevOps Workshop – 2

By Zama

Agenda

- Storage overview
- Volume Types
- Docker compose
- Hands-on

Docker: Volume

There are three types of volumes: *host*, *anonymous*, and *named:*

- A **host volume** lives on the Docker host's filesystem and can be accessed from within the container. To create a host volume:

```
docker run -v /path/on/host:/path/in/container ...
```

- An **anonymous volume** is useful for when you would rather have Docker handle where the files are stored. It can be difficult, however, to refer to the same volume over time when it is an anonymous volumes. To create an anonymous volume:

```
docker run -v /path/in/container ...
```

- A **named volume** is similar to an anonymous volume. Docker manages where on disk the volume is created, but you give it a volume name. To create a named volume:

```
docker volume create somevolumename
docker run -v name:/path/in/container ...
```

Docker: Copy files manually

Create folder and file:

```
mkdir data
nano cp.sh
    echo '<EMP_ID> Running cp.sh file - '$(date)
```

CP:

Copy file from host to container

```
docker cp data/cp.sh /java8-app-c1:/data
```

Copy file from container to host

```
docker cp java8-app-c1:/data/cp.sh cp_2.sh
```

Verify:

```
//Copy file from host machine to container
    docker cp data/ java8-app-c1:/data

//Login to the docker container
    docker exec -it java8-app-c1

//Verify the file
    ls -l data
    cat data/cp.sh

//Run file
    sh data/cp.sh
```

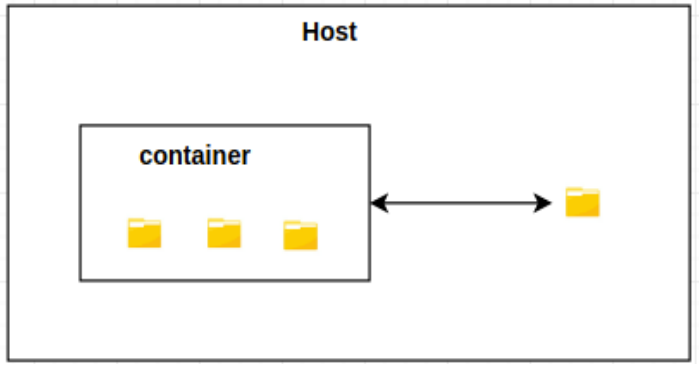
Docker: COPY ***vs*** ADD

COPY:

Copy files/folder from host machine to container without any conversions.

Add this line in Dockerfile:

COPY *data/copy_1.tar.xz /*



ADD:

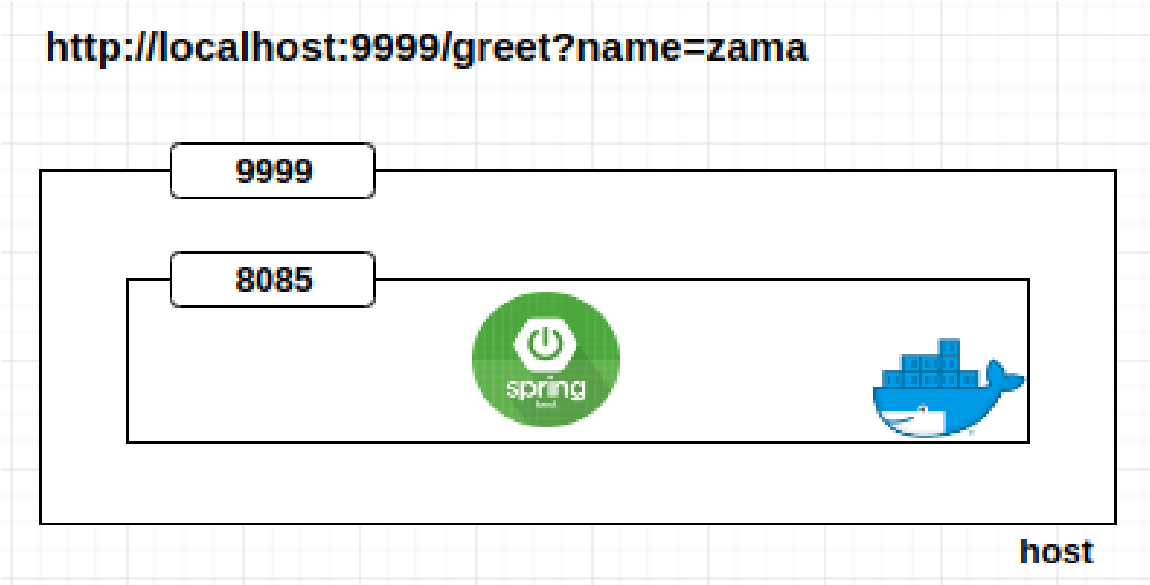
Allows <src> to be a URL (If is a local tar archive in a recognized compression format (identity, gzip, bzip2 or xz) then it is unpacked as a directory).

Add this line in Dockerfile:

ADD data/add_1.tar.xz /

EvokeTechnologies

Port Mapping



Hands-on: Assignment 2 – Spring boot

- Run the spring boot app without docker
- Dockerize the spring boot app
- Build docker image
- Run the app with docker
- Push the image to Docker Hub (Public docker registry)
- Verify in another machine

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

Basic 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
```

Sample YAML file

docker-compose.yml

```
version: '3'

services:
   api:
     image: spring-boot-demo
     ports:
       - “9999:8085”
```

Resources

- Instructions and source code for the hands-on is available at this location: https://github.com/mbzama/docker-training

Thank You

EvokeTechnologies