DAAD Summer School 2019

ulm university universität ulm

Tools & Concepts for Cloud Deployments

Exercise 5: Containers

SummerSchool 2019, Curitiba

Christopher B. Hauser

Institute of Information Resource Management, Ulm University

Section 1

Exercise 5: Containers

Overview

Welcome to exercise 5. This time we will have the following lessons:

1. Concept of Containers
2. Mediawiki with Docker

Lessons learned

- Learn about Containers and their relation to virtual machines
- How to install and use LXC containers
- How to install and use Docker containers
- Multi-tier applications like Mediawiki with Docker and Dockerfile
- Docker-compose for single-host orchestration

Section 2

Answers to questions

Lesson 1: Concept of Containers

Question: Containers, LXC and Docker

What architectural implications are required for an application to run in containers?

- Service-oriented: Containerized applications work best when implementing a service-oriented design. Service-oriented applications break the functionality of a system into discrete components that communicate with each other over well-defined interfaces. One container ideally represents a single service only.
- Application state: The containers should be stateless and store the application state somewhere outside the containers (e.g. in mounted volumes). Having stateless containers allow to horizontally scale by design, and enable a fast error fail over by re-creating failed containers.

Lesson 2: Mediawiki with Docker

Questions: Experiences with Docker

Practically, where do you see benefits and drawbacks in the use of virtual machines versus Docker containers?

Docker containers are rather application component centric, virtual machines are operating system centric. Virtual machines are more flexible, when the purpose is unclear (e.g. not specified in a script or Dockerfile). Docker containers on the other hand are very lightweight and fast. While creating a new virtual machine can take up to some minutes, a Docker container is up within seconds.

Why are the two layers "Cloud Platform" and "Virtual Resources" still necessary although we have containers? Or why are both layers not necessary when working with containers?

A strong isolation between virtual machines is necessary to share the physical hardware with multiple users. While

Question: Docker distributed

Can you find a solution to distribute Docker containers on multiple hosts?

So called Orchestrators are needed. Some examples are: Docker Swarm, Google's Kubernetes, Apache Mesos, Rancher

These orchestrators schedule the containers in available resources - virtual machines or bare metal servers.

Section 3

Solution for practical part

Mediawiki with Docker

You should have the following Docker images:

```
REPOSITORY                TAG       IMAGE ID   CREATED
clouds/loadbalancer       latest    ...        2 hours a
clouds/mediawiki          latest    ...        2 hours a
clouds/database           latest    ...        3 hours a
telegraf                  latest    ...        3 days ag
ubuntu                    14.04     ...        3 days ag
nginx                     latest    ...        6 days ag
chronograf                latest    ...        2 weeks a
influxdb                  latest    ...        2 weeks a
mariadb                   5.5       ...        3 weeks a
```

And you should have the following Docker containers:

```
ID  IMAGE                     ...  STATUS          PORTS
... telegraf                  ...  Up 7 seconds    ...
... chronograf                ...  Up 8 seconds    0.0.0.
... influxdb                  ...  Up 9 seconds    8086/t
```

Mediawiki with Docker Compose

To install docker compose, download the binary and make it executable:

```
sudo -s
curl -L \
 https://github.com/docker/compose/releases/downloa
 > /usr/local/bin/docker-compose
chmod +x /usr/local/bin/docker-compose
```

The working `docker-compose.yaml`:

```
version: '2'
services:
    web1:
        build: Mediawiki
        image: clouds/mediawiki
    web2:
        build: Mediawiki
        image: clouds/mediawiki
```