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?
- What are pros/cons of containers compared to virtual machines?

TODO: More questions here?

Question: LXC Containers

- Where are more processes running: on the vm operating system or inside the container? And why?
- How does the networking look like? Why is it a good idea to have a private IP for each containers?
- What resources can be limited via `lxc config set [container] limits.*` command?

Question: Docker and Docker Hub

Now that you are familiar with the basics of Docker:

- What are the differences between a Dockerfile and a Docker image? Can you imagine pros/cons?
- How does a typical workflow for deploying a new application component look like?

Have a look at the Docker Hub [1].

- Do you think it was useful to create an image for Apache by ourselves?
- How are images in Docker Hub created and maintained?

[1] https://hub.docker.com/

Lesson 2: Mediawiki with Docker

Questions: Experiences with Docker

Since the beginning of the exercises you made experiences with virtual machines on OpenStack, automated resource allocation with Terraform and automated application deployment with cloud-init. Finally, we just 'dockered' the Mediawiki example.

- Practically, where do you see benefits and drawbacks in the use of virtual machines versus Docker containers? (E.g. creation time, image size, descriptiveness, ...)
- Looking at the cloud stack from the beginning of this exercise, 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?

Question: Docker distributed

So far we used one virtual machine to host all the Docker containers. This will of course not scale eventually.

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

Solution for practical part

docker-compose.yaml:

```
version: '2'
services:
    web1:
        build: Mediawiki
        image: clouds/mediawiki
    web2:
        build: Mediawiki
        image: clouds/mediawiki
    database:
        build: Database
        image: clouds/database
    loadbalancer:
        build: Loadbalancer
        image: clouds/loadbalancer
        ports:
         - 80:80
    influxdb:
        image: influxdb
    chronograf:
        image: chronograf
        ports:
         - 8888:8888
    telegraf:
        image: telegraf
        volumes:
         - /var/run/docker.sock:/var/run/docker.sock:ro
         - /sys:/rootfs/sys:ro
```

```
      - /proc:/rootfs/proc:ro
      - /etc:/rootfs/etc:ro
      - ./Monitoring/telegraf.conf:/etc/telegraf/telegraf.conf:ro
    environment:
      - HOST_PROC=/rootfs/proc
      - HOST_SYS=/rootfs/sys
      - HOST_ETC=/rootfs/etc
```

Next, build the containers and start the services

```
docker-compose build
docker-compose up
```