UniLaSalle Amiens

DEVOPS TOOLS (IAC)

LABS

SECOND SESSION (EXAM)
USE ANSIBLE AND TERRAFORM

UniLaSalle
Amiens

Install or make sure that VirtualBox is usable

Use this link to install VirtualBox

Install Vagrant

Install Vagrant and make sure it is usable using Powershell and typing `vagrant --version`

Create a directory containing all future labs

Name it `ansible-tp`

Create a file `vagrantfile` containing following (sent by mail)

Start your environment

```
vagrant up
```

Connect to iac-control machine

```
vagrant ssh iac-control
```

Refer to the doc to install Ansible

Install ansible using this link, refer **ONLY** to `Install Ansible` and `Confirming your installation`

Make sure you can use Ansible *(you may need to adapt you PATH environment variable)* → `export PATH=$PATH:[...]`

LAB 2 INSTALL TERRAFORM & DOCKER

On the VM `iac-control`

Install Terraform

Then install Docker & do post-install to add user vagrant to group docker

Make sure both are well installed

Using Terraform and following provider

- Create 2 containers based on following image: https://hub.docker.com/r/takeyamajp/ubuntu-sshd:
  - One named `database` (Port forward 22 to 2200 in terraform)
  - Another named `webserver` (Port forward should be 22 to 2201 in terraform)
- Create a network "unilasalle", containers should use it Following resources should be used: docker_network, docker_container & docker_image

On iac-control, generate a SSH key using ssh-keygen, then copy key to each server (dbserver & webserver)

Container root password is **root**

Bonus

Instead of manually using ssh-copy-id, use terraform to run the command example

LAB 4 CREATE ANSIBLE INVENTORY

Create an Ansible inventory, make sure containers are accessible, use one of the way below:

- First one:
  - Use ansible_host, ansible_port and ansible_user in inventory to force ansible IP, port and user
- Second one:
  - Edit /etc/hosts based on containers IP (accessible using docker inspect command) Then, use ansible modules to contact your containers
- Use module `ping`
- Create two groups: web & db

LAB 5 DELIVER AN SSH BANNER

On each server, deliver an SSH banner adding a warning: **Authorized access only, server belongs to Unilasalle** (example)

Tips:

- Use module `copy` & `lineinfile`
- Use /etc/banner as filename
- Use `docker stop` and `docker start` to check if everything works well, **do not restart service**

Bonus

Find a way to restart container using Ansible and Handlers

On all containers

Instead of delivering a static file, make use of ansible magic variables and deliver a file based on these variables:

- Hostname
- IP
- Groups

On all containers

Use module group to create two groups:

- supervision
- applicative

Use module users to create two users

- applicative (belonging to group applicative)
- supervision (belonging to group supervision)

On server webserver

Create user:

- web (belonging to applicative)

On server dbserver

Create user:

- dbadmin (belonging to applicative)

LAB 8 INSTALL WEBSERVER

- Install apache2 on webserver using ansible
- Install php on webserver with postgresql drivers (php-pgsql)
- Make sure server is UP and PHP is UP also (tips: curl on the container)
- Make sure apache2 starts at boot

- Install postgresql
- Change listening port from 5432 to 5431 (use handlers to restart if there are any change, use module lineinfile)

Bonus

Try to automate database creation:

- Create a database named Unilasalle
- Create a user named unilasalle-admin with all privileges on database Unilasalle

LAB 10 - BONUS DEPLOY SMALL PHP PAGE

- Write a small PHP page allowing you to connect to the database
- Deliver it to webserver
- Make sure database connection is OK

LAB 11 - BONUS SETUP A LOADBALANCER

- Deploy a second web server
- Deploy a loadbalancer
- Install haproxy and redirect traffic on webserver