UniLaSalle
Amiens

DEVOPS TOOLS (IAC)

LABS

FIRST SESSION
DISCOVER ANSIBLE

UniLaSalle Amiens

SUMMARY

UniLaSalle Amiens

SETUP INSTALL YOUR ENVIRONMENT

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 ansible-control machine

```
vagrant ssh ansible-control
```

LAB 1 INSTALL ANSIBLE

Refer to the doc to install Ansible

Install ansible using this link Make sure you can use Ansible *(you may need to adapt you PATH environment variable)*

Modify /etc/hosts file

It will allow you to contact hosts using name instead of IP, add it to end of file

```
192.168.56.101 db01 db01

192.168.56.102 web01 web01

192.168.56.103 web02 web02

192.168.56.104 loadbalancer
```

LAB 1 INSTALL ANSIBLE

Create an SSH key

```
ssh-keygen
```

Leave all value to default one (press enter)

Deploy SSH key on all servers

```
ssh-copy-id web01
ssh-copy-id web02
ssh-copy-id db01
ssh-copy-id loadbalancer
```

Password for each server is **vagrant**

LAB 2 CREATE AN INVENTORY AND MAKE SURE HOSTS ARE REACHABLE

Using class slides and documentation, create an inventory

Documentation

Create a folder `TP` and create a file named `inventory` into it.

You should use this file as inventory. *Ask if you need tips or help*

Use some ad-hoc commands to reach your servers

For example:

```
ansible all -i [name of inventory file you created] -m [name of a test module such as ping, setup]
```

LAB 3 RETRIEVE HOSTS FACTS

A special module allow you to retrieve a lot of data about host

Search for this module and look at all the data you can retrieve

List some which can be useful according to you

Ask for tips of help

Call me to validate this step and make sure you understand all of this part

LAB 4 USE ADHOC MODULES AGAINST SERVERS

- Align inventory:
  - Create 3 groups: `webservers`, `dbservers` and `loadbalancers`
  - Put each VMs into a group
- Using adhoc modules, install this package: `apache2` on `webservers` group
- Do the same with `dbservers` group and install `mariadb`
- Use ansible adhoc to make sure both services are started and enabled (service should start at boot, to make sure, use module reboot on all servers)
- Re-use the commands you issued above and make sure there are no changes (IDEMPOTENCY)

LAB 5 CREATE A SIMPLE PLAYBOOK

Convert all you have done before to a playbook.

- In addition, find a way to change default apache2 page (we should be able to access web server using their IP addresses)

LAB 6 USE JINJA2 TEMPLATING FOR MOTD

On all servers

- Create a Jinja2 template to serve as a MOTD, form is free but it should include at least:
  - Server name
  - Ansible groups to which this system belongs to
  - Current date
  - Main IP address of the server

On web servers

- Create a Jinja2 template deploying a HTML page containing
  - Group names
  - Linux kernel version
  - Server is virtual ?

LAB 7 USE VARIABLES

- You should probably move your inventory to something else *(create a ./inventory folder, move your file as hosts into ./inventory folder)*
  - Create a folder named `host_vars` and two files `web01` and `web02`
- Create a variable named `web_package`
  - On host `web01` it should have value `apache2`
  - On host `web02` it should have value `nginx`
- Adapt your playbook to make sure this variable is used to install the package (and start it also)

LAB 8 CREATE A ROLE

For database servers group, you should create a role using:

- A `roles` folder
- Run following command to create a role (inside roles folder)

```
ansible-galaxy init database
```

- Your role should
  - Install `mariadb`
  - Start and enable it
  - Create a table named `ansible-tp`
  - Create a user named `YOUR-NAME` with a password and having full rights on table `ansible-tp`
- All variables should be customizable (starting with `database_`), for example

```
database_user
```

LAB 8 CREATE A ROLE

Do the same with a webserver role:

- Your role should
  - Install `nginx` or `apache2` based on a variable
  - Deploy a default page different for each server (you should use when condition)
  - Install a template containing all the informations you want to show
- All variables should be customizable (starting with `webserver_`), for example

  `webserver_package`

LAB 9 USE HANDLERS

Edit database role

You should now bind service to port **3307** instead of **3306**.

After modifying the conf, call a handler **restarting the service** and **make sure service listen on port 3307**.