TD03 - Ansible

Prerequisites

Till now we have only been preparing our applications to be deployed. However we did not deploy anything. That's where Ansible takes place. Ansible is basically a tool to manage your servers, provision them and deploy your applications on them. This is not the only solution on the market, you'll hear also about Chef, Puppet, Terraform during your developper life. All of them have their advantages and disadvantages, it is up to you to play with and make your own decisions.

This introduction will be pretty fast, it is just here to make you manipulate the tool a little bit with some simple ad hoc commands. You will go deeper into the practical part.

Do you Ansible?

```
$ ansible --version
```

Check your installed version, config file location, associated python version and more. If you do not ansible, head to: docs.ansible.com

SSH remote connection

Each of you has normally received a server domain name that should be yourname-yourlastname-formation.takima.io and a private key to SSH to it.

This server is yours, you will be the only one to manipulate it. In order to play with it, you can simply ssh to it.

SSH basically means Secure Shell, it is both a software and a communication protocol that uses the protocol/port TCP/22 of your machines to communicate. It is called Secure because the communication is encrypted using your ssh key pair.

Before trying any command, you should know that your private key requires restricted permissions to be used. Change the rights of your key:

```
$ chmod 400 <path_to_your_key>
```

Now your key can be used to ssh to your server. Go on and hit :

```
$ ssh -i <path_to_your_key> centos@<your_server_domain_name>
```

Why do we have to add this "centos@" ? Your machines run under a CentOS distribution, and the default user is centos, this is why we specify which user we want to use.

Now you are connected to your instance, nothing important to see here. You can exit whenever you want using the command:

```
$ exit
```

Why do we show you how to make a remote SSH connection ? Because it is basically what does Ansible to communicate with your server. Now as you can already guess, Ansible will require some configurations to be able to access your machine.

Say Hello from Ansible

We will simply use a ping command from Ansible to say hello to our server. Actually, the Ansible ping command does a bit more than just the usual bash ping command. If Ansible responds a "pong" to you, it means that your server is available, that the user provided exists and that Ansible was able to authenticate to your server. In summary, it tells you that your Ansible configuration works.

First, we need to add our server name to our Ansible hosts list:

```
$ vim /etc/ansible/hosts
```

Add your server domain name into the file and save it. Now Ansible knows you remote host. Now let's hit:

```
$ ansible all -m ping
```

And it… doesn't work. Why? Because Ansible has been access denied as it did not provide either a user nor a private ssh key. Now try again with this one:

```
$ ansible all -m ping --private-key=<path_to_your_ssh_key> -u centos
```

And now it should respond "pong", which means you are successful with your configurations.

Setup an Apache Server

Now that we are able to access our instance with Ansible, let's see how powerful Ansible can be for provisioning your web server.

We are going to ask Ansible to install Apache into your instance to make it a webserver.

```
$ ansible all -m yum -a "name=httpd state=present"
--private-key=<path_to_your_ssh_key> -u centos
```

And it… doesn't work ! Actually, like in every system, you need to be root in order to install a software. Fortunately, Ansible can take care of this. Try again with the following command:

```
$ ansible all -m yum -a "name=httpd state=present"
--private-key=<path_to_your_ssh_key> -u centos --become
```

The --become flag tells Ansible to perform the command as a super user. Keep in mind that the centos user is part of the wheel group which is the CentOS super users group. It would not be possible using a normal user.

Now you have successfully installed Apache on your server. We will go on and create an html page for our website:

```
$ ansible all -m shell -a 'echo "<html><h1>Hello CPE</h1></html>" >>
/var/www/html/index.html' --private-key=<path_to_your_ssh_key> -u centos
--become
```

Now start your Apache service:

```
$ ansible all -m service -a "name=httpd state=started"
--private-key=<path_to_your_ssh_key> -u centos --become
```

Now if you try to connect to your instance on your browser you'll see… Nothing. The connection hangs, but nothing happens. This is because your server doesn't allow any transport to the port 80. You need to open this port:

```
$ ansible all -m firewalld -a "port=80/tcp permanent=true state=enabled
immediate=true" --private-key=<path_to_your_ssh_key> -u centos --become
```

Connect again to your server from your browser and… it works ! Well done you've set up your very first server with Ansible. Now move on to the practical part.