UNIVERSIDAD DE HUELVA

MASTER EN INGENIERÍA EN INFORMÁTICA



Cátedra: Cloud Computing
Infraestructura como Código

Ciclo lectivo 2022-2023

Profesor: José Luis Álvarez Macías

Alumno: Monastyrski, Carlos Alberto

Infraestructura como Código

Instala Ansible en una máquina de tu elección

Utilicé pip, ya que no tenía la posibilidad de utilizar sudo:

```
Counse which the control of the cont
```

Crea dos máquinas virtuales utilizando un Playbook para crear las máquinas con Ansible

Fue necesario configurar los permisos y generar un archivo JSON para que en el playbook se pudiera autenticar ansible al realizar las peticiones:

```
$ gcloud config set project smart-howl-365613
$ gcloud iam service-accounts create autenticacion
$ gcloud projects add-iam-policy-binding smart-howl-365613
--member
"serviceAccount:1003776626205-compute@developer.gserviceaccount.co
m" --role "roles/editor"
$ gcloud iam service-accounts keys create permisos --iam-account
1003776626205-compute@developer.gserviceaccount.com
```

Una vez realizado esto, cree mi Playbook infra.yml:

```
- name: 'Deploy gcp vm'
hosts: localhost
connection: local
become: false
gather_facts: no
```

```
vars:
```

```
name: create a disk 2
```

```
- name: external-nat
    nat_ip: "{{ address2 }}"
    type: ONE_TO_ONE_NAT
    state: present
    auth_kind: "{{ auth_kind }}"
    service_account_file: "{{ service_account_file }}"
```

Y lo ejecute con ansible-playbook infra.yml:

```
charly pomestyrak(cloudshell: (mart-base)-48643) ansible-playbox infra,ymi
(MARTHO) (No investory was praced, only implical localizate in variable. Note that the implicit localizate is variable. Note that the implicit localizate and was a solicate and was a solicate.

DAT (make a solicate)

TAT (make a solicate)

TAT (make a solicate)

TAT (make a solicate)

TAT (make a instance)

TAT
```

Y quedaron creadas ambas instancias:



Crea un inventario con esas máquinas

nano inventory.ini

```
[server]
35.195.6.122
[db]
104.155.28.188
```

Crea un Playbook para hacer ping a ambas máquinas

En este paso, antes tenemos que habilitar la comunicación mediante SSH entre las máquinas, añadiendo la clave ssh que se generó en mi terminal de cloud en ~/.ssh/authorized keys de ambas instancias.

Y con esto creamos el playbook:

```
- name: Ping a la db
hosts: db
```

```
gather_facts: false
tasks:
    - name: Realizar un ping a la db
    ping:
- name: Ping al server
hosts: server
gather_facts: false
tasks:
    - name: Realizar un ping al server
ping:
```

Y lo ejecutamos con ansible-playbook -i inventory.ini ping.yml

Crea un Playbook para instalar Apache y PHP, en webserver, y MySQL en database

El playbook que cree se llamó: install-features.yml:

```
state: present

- hosts: db
become: true

tasks:
- name: Actualizar apt-get
apt:
    update_cache: yes

- name: Instalar MySQL
apt:
    name: mysql-server
    state: present
```

Y lo ejecuté con: ansible-playbook -i inventory.ini install-features.yml

```
Charly_monastyrski@cloudshell:- (agile-alignment-384815) % ansible-playbook -i inventory.ini install-features.yml

PLAY (Server)

TASK (Gathering Facts)

Out (Distribution)

TASK (Instaliar apt-get)
changed: [35.195.6.122]

TASK (Instaliar Apache)
changed: [35.195.6.122]

TASK (Instaliar PRP)
changed: [35.195.6.122]

TASK (Instaliar PRP)
changed: [35.195.6.128]

TASK (Gathering Facts)

Out (US/155.26.188)

TASK (Instaliar MySGL)
changed: [104.155.26.188]

PLAY RECAP

104.155.20.188 : bb-1 changed-2 unracchable=0 failed-0 skipped-0 rescued-0 ignored-0

35.195.6.122 : bb-4 changed-2 unracchable=0 failed-0 skipped-0 rescued-0 ignored-0

35.195.6.122 : bb-4 changed-2 unracchable=0 failed-0 skipped-0 rescued-0 ignored-0

35.195.6.122 : bb-4 changed-2 unracchable=0 failed-0 skipped-0 rescued-0 ignored-0

35.195.6.122 : bb-4 changed-3 unracchable=0 failed-0 skipped-0 rescued-0 ignored-0

Skipped-0 rescued-0 ignored-0

Skipped-0 rescued-0 ignored-0
```

Fue necesario antes de ejecutar esta línea modificar en las instancias para que no solicite la clave de sudo para ejecutar los comandos:

sudo visudo

```
GNU nano 4.8 /etc/sudoers.tmp

This file MUST be edited with the 'visudo' command as root.

# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.

# See the man page for details on how to write a sudoers file.

# Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/snap/bin"

# Host alias specification

# User alias specification

# User privilege specification

# User privilege specification

# Wembers of the admin group may gain root privileges
% admin ALL=(ALL) NOPASSWD:ALL

# Allow members of group sudo to execute any command
% sudo ALL=(ALL:ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:
# includedir /etc/sudoers.d
charly_monastyrski_ALL=(ALL) NOPASSWD: ALL
```

Y al final del archivo se agrega el nombre del usuario de nuestra terminal seguido por ALL=(ALL) NOPASSWD: ALL