

SP16-M4 ~ Réalisation des différents script ansible

Table des matières

SP16-M4 ~ Réalisation des différents script ansible
Ajout du serveur web dans l'inventaire
Installation de nginx
Création de la template de nginx
Création du playbook pour l'installation de nginx
Création d'un playbook pour la synchronisation du siteweb
Création d'un site web pour la mairie
Exécution des playbooks

1. Ajout du serveur web dans l'inventaire

1. Il faut d'abord transférer notre clé public ssh sur le serveur.

```
ssh-copy-id gitlabadmin@172.16.56.210
```

2. Mot de passe du root

Afin d'éviter l'erreur `sudo: il est nécessaire de saisir un mot de passe`, il faut retirer les paramètres du root dans `/etc/sudoers`.

```
- $sudo ALL=(ALL:ALL) ALL
+ $sudo ALL=(ALL:ALL) NOPASSWD: ALL
```

3. Nous devons ajouter le serveur dans le fichier d'inventaire `inventories/production/hosts`.

```
+ [serveur_web]
+ 172.16.56.210 ansible_connection=ssh ansible_ssh_user=webadmin
```

4. Ajout des variable du serveur_web `inventories/production/host_vars/serveur_web` .

```
domain: sca.fr
```

5. Nous pouvons enfin tester la connexion vers le serveur web

```
ansible -i inventories/production/hosts -m ping serveur_gitlab
```

```
172.16.56.210 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

2. Installation de nginx

2.1. Création de la template de nginx

Le fichier de configuration nginx seront stocker dans une template `templates/site.conf.j2` .

C'est un fichier jinja qui utilise des variable afin de crée des fichier de configuration unique par serveur.

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/{{ domain }};
    location / {
        try_files $uri $uri/ =404;
    }
}
```

2.2. Création du playbook pour l'installation de nginx

Nous allons ensuite crée le playbook `nginx.yml` qui va nous paramètre d'installer.

```
- hosts: serveur_web
  become: yes
  tasks:
    - name: "apt-get update"
      apt:
        update_cache: yes
        cache_valid_time: 3600

    - name: "install nginx"
      apt:
        name: ['nginx']
        state: latest

    - name: "create www directory"
      file:
        path: /var/www/{{ domain }}
        state: directory
        mode: '0775'
        owner: "{{ ansible_user }}"
        group: "{{ ansible_user }}"

    - name: delete default nginx site
      file:
        path: /etc/nginx/sites-enabled/default
        state: absent
      notify: restart nginx

    - name: copy nginx site.conf
      template:
        src: site.conf.j2
        dest: /etc/nginx/sites-enabled/{{ domain }}
        owner: root
        group: root
        mode: '0644'
      notify: restart nginx

  handlers:
    - name: restart nginx
      service:
        name: nginx
        state: restarted
```

2.3. Création d'un playbook pour la synchronisation du siteweb

Nous allons crée un playbook `sync.yml` qui va transférer le dossier `site` vers le serveur web.

```
- hosts: web
tasks:
- name: "sync website"
  synchronize:
    src: site/
    dest: /var/www/{{ domain }}
    archive: no
    checksum: yes
    recursive: yes
    delete: yes
    become: no
```

2.4. Création d'un site web pour la mairie

Enfin nous allons crée le site web de la mairie. `site/index.html`

```
<!DOCTYPE html>
<html lang="en">
<title>Mairie - saint chély d'apcher</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<header>
  <h1>Mairie de Saint Chély d'Apcher</h1>
</header>

<body>
  <p>Bienvenue sur le serveur de la marie</p>
</body>

</html>
```

3. Exécution des playbooks

Nous pouvons enfin exécuter le playbook nginx pour installer le logiciel.

```
ansible-playbook -i inventories/production nginx.yml
```

```
PLAY [serveur_web]

TASK [Gathering Facts]
ok: [172.16.56.210]

TASK [apt-get update]
changed: [172.16.56.210]

TASK [install nginx]
changed: [172.16.56.210]

TASK [install rsync]
changed: [172.16.56.210]

TASK [create www directory]
changed: [172.16.56.210]

TASK [delete default nginx site]
changed: [172.16.56.210]

TASK [copy nginx site.conf]
changed: [172.16.56.210]

RUNNING HANDLER [restart nginx]
changed: [172.16.56.210]

PLAY RECAP
172.16.56.210      : ok=8    changed=7    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

Enfin nous allons transférer le site via rsync.

```
ansible-playbook -i inventories/production sync.yml
```

```
PLAY [serveur_web]
```

```
TASK [Gathering Facts]
```

```
ok: [172.16.56.210]
```

```
TASK [sync website]
```

```
changed: [172.16.56.210]
```

```
PLAY RECAP
```

```
172.16.56.210      : ok=2    changed=1    unreachable=0    failed=0  
skipped=0    rescued=0    ignored=0
```