### Travail pratique 4 – Ansible, installation d’un service

Informations

**Évaluation** : 20 % de la session

**Type de travail** : individuel

**Date de remise** : 9 février 2021, sur léa

**Durée** : 4 heures

**Système d’exploitation** : Ubuntu

**Environnement** : virtuel

Objectifs

Ce travail pratique a pour objectifs :

* Configurer Ansible.
* Comprendre Ansible.
* Vérifier les communications avec le serveur Web local.
* Créer des Playbooks Ansible pour automatiser l'installation du serveur Web.
* Ajouter des options à votre Playbook Ansible pour serveurs Web Apache.

Dans ce travail pratique, vous allez d'abord configurer Ansible afin qu'il puisse communiquer avec une application de serveur web. Vous allez ensuite créer un playbook qui automatisera le processus d'installation d'Apache sur le serveur web. Vous allez également créer un playbook personnalisé qui installe Apache avec des instructions spécifiques.

## Configurer Ansible

Dans cette partie, vous allez configurer Ansible pour utiliser l'adresse IPv4 loop1 pour un serveur Web local.

### Vérifiez le serveur SSH.

Le serveur SSH est normalement activé dans votre VM, nous allons vérifier s’il est actif. Sinon, nous allons le lancer. Vérifiez-le et s’il n’est pas actif, démarrez-le.

devasc@labvm:~$ **sudo systemctl status ssh**

devasc@labvm:~$ **sudo systemctl start ssh**

Dans Ansible, pour pouvoir utiliser une connexion ssh avec le mot de passe défini dans le fichier **hosts**, vous devez installer **sshpass** :

**Question -** Quelle est la commande pour installer sshpass ?

**Installer sshpass**

### Ouvrez le répertoire ansible dans VS Code.

* + - 1. Ouvrez **VS Code**. Vous pouvez également utiliser un autre éditeur de code.
      2. Cliquez sur **File > Open Folder...** et naviguez jusqu'au dossier **~/ansible**
      3. Cliquez sur **OK**.
      4. Le sous-répertoire pour l’exercice est maintenant chargé dans le volet VS Code **EXPLORER** pour votre commodité. Dans ce travail, vous travaillerez avec le répertoire **local**.

### Créez un fichier hosts Ansible

* + - 1. Créez un fichier **hosts** dans le répertoire **local**.
      2. Ajoutez les lignes suivantes au fichier **hosts** et enregistrez.

[webservers]

192.168.20.250 ansible\_ssh\_user=*VotreNomLogin* ansible\_ssh\_pass=*VotreMotDePasse*

*///// ajouter le mot de passe*

*Sudo pass ansible\_sudo\_pass=Passw0rd*

* + - 1. Les informations d'identification que vous utilisez sont les informations d'identification pour votre machine virtuelle. L'adresse IPv4 que vous utiliserez pour cet exercice est 192.168.20.250. Il s'agit de l’adresse IPv4 statique sur la machine virtuelle sous l'interface **loop1**, comme indiqué dans la sortie de la commande **ip addr**.

❯ ip addr

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: ens160: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP group default qlen 1000

link/ether 00:50:56:9a:d2:da brd ff:ff:ff:ff:ff:ff

altname enp3s0

inet 10.100.1.199/24 brd 10.100.1.255 scope global dynamic ens160

valid\_lft 10339sec preferred\_lft 10339sec

inet6 fe80::250:56ff:fe9a:d2da/64 scope link

valid\_lft forever preferred\_lft forever

3: loop1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000

link/ether fe:31:03:55:be:7f brd ff:ff:ff:ff:ff:ff

inet 192.168.20.250/32 scope global loop1

valid\_lft forever preferred\_lft forever

inet6 fe80::fc31:3ff:fe55:be7f/64 scope link

valid\_lft forever preferred\_lft forever

4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default

link/ether 02:42:02:d0:f2:78 brd ff:ff:ff:ff:ff:ff

inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0

valid\_lft forever preferred\_lft forever

### Créez un fichier ansible.cfg.

* + - 1. Nous devons configurer notre environnement local ansible avec des proriétés globales spécifiques pour nous permettre d’éxécuter des commandes ansible. Lorsque nous exécutons des commandes ansible, le fichier de configuration sera recherché dans cette ordre.
         1. La variable d’environnement ANSIBLE\_CONFIG. Naturellement, s’elle est configurée.
         2. Le fichier ansible.cfg dans le répertoire courant.
         3. Le fichier ~/.ansible.cfg (dans le répertoire home).
         4. Finalement, dans /etc/ansible/ansible.cfg.
      2. Dans le sous-répertoire **local**, créez un fichier **ansible.cfg**.
      3. Ajoutez les lignes suivantes au fichier et enregistrez-le. Le fichier **ansible.cfg** indique à Ansible où trouver le fichier d'inventaire et définit certains paramètres par défaut.

[defaults]

# Use local hosts file in this folder

inventory=./hosts

# Don't worry about RSA Fingerprints

host\_key\_checking = False

# Do not create retry files

retry\_files\_enabled = False

## Vérifier les communications avec le serveur Web local

Dans cette partie, vous vérifierez qu'Ansible peut envoyer des commandes au serveur Web local.

### Utilisez le module ping pour vérifier qu'Ansible peut effectuer un ping sur le serveur Web.

L’utilisation de commandes ansible ad hoc est sensiblement identique à l’utilisation d’une commande bash.

La syntaxe d’une commande ansible est :

ansible <HOST> -b -m <MODULE> -a "<ARG1 ARG2 ARGN>" -f <NUM\_FORKS>

* + - * + HOST est un hôte ou un groupe d'hôtes défini dans le fichier d'inventaire Ansible.
        + -b est pour ‘become’ (ceci remplace l’indicateur -s déprécié comme dans «sudo»).

Ansible augmente l'autorisation à --become-user en utilisant la méthode définie par --become-method.

Par défaut become-user est root.

Par défaut become-method est sudo.

* + - * + -m est pour le module ou la commande à utiliser.
        + -a est pour que les paramètres a passer. S'il est utilisé sans module, c'est comme exécuter une commande shell sur le(s) système(s) cible(s).
        + -f est utilisé pour définir des fourches pour le parallélisme, ce qui permet à Ansible d'exécuter des plays simultanément sur de nombreux hôtes.

Les modules sont des unités de code discrètes qui peuvent être utilisées à partir de la ligne de commande ou dans une tâche de playbook.

* + - * + Les modules peuvent accepter des arguments.
        + La plupart des modules prennent des arguments **key=valeur**, délimités par des espaces.
        + Il y a quelques exceptions trouvées dans les modules de commande et de shell.

Modules communs :

Module: ping

Objectif: vérifier la connectivité Ansible entre les hôtes.

Arguments: aucun

Module: setup

Objectif: recueillir des faits Ansible.

Arguments: aucun

Module: service

Objectif: contrôler les démons de service.

Arguments: name=<SERVICE\_NAME> state=<STATE>

Module: copy

Objectif: copier un fichier vers un emplacement particulier sur un hôte cible.

Arguments: src=<SOURCE\_PATH> dest=ABSOLUTE\_DESTINATION\_PATH>

Utilisez le module Ansible **ping** pour vérifier les communications avec les périphériques répertoriés dans le groupe de **webservers** de votre fichier d'inventaire **hosts**.

~/ansible/local$ **ansible webservers -m ping**

Consultez le site web de référence d’Ansible (<https://docs.ansible.com/ansible/latest/index.html>) et faire une recherche pour « module ping ».

**Question -** Quelle réponse êtes-vous censé recevoir ?

Si plusieurs périphériques étaient répertoriés sous le groupe de **webservers** dans votre fichier d'inventaire  **hosts**, la sortie indiquerait des informations similaires pour chaque périphérique.

Si vous n’avez pas reçu de succès, vérifer vos fichiers **ansible.cfg** et **hosts**.

### Utilisez le module de commande pour vérifier qu'Ansible peut communiquer avec le serveur Web.

Utilisez le module de **commande** Ansible pour vérifier les communications avec les périphériques répertoriés dans le groupe **webservers** de votre fichier d'inventaire **hosts**. Dans cet exemple, vous envoyez l'argument ‑**a "/bin/echo hello world"** pour demander au serveur web local de répondre (**a**nswer) avec "hello world".

~/ansible/local$ **ansible webservers -m command -a "/bin/echo hello world"**

192.168.20.250 | CHANGED | rc=0 >>

hello world

~/ansible/local$

## Créer des Playbooks Ansible pour automatiser l'installation du serveur Web

Dans cette partie, vous allez créer deux playbooks Ansible. Le premier playbook va automatiser le test d'écho que vous avez fait dans la partie précédente. Imaginez que vous apportez une centaine de serveurs Web en ligne. Le groupe [webservers] dans le fichier **hosts** répertorie toutes les informations nécessaires pour chaque serveur web. Vous pouvez ensuite utiliser un simple playbook pour vérifier les communications avec chacun d'eux à l'aide d'une seule commande. Dans le second playbook, vous allez créer et automatiser l'installation du logiciel serveur web Apache.

### Créez votre playbook Ansible pour tester votre groupe de webservers.

Dans cette étape, vous allez créer un playbook Ansible pour exécuter la même commande **echo**.

* + - 1. Dans VS Code, créez un nouveau fichier dans le répertoire **local** avec le nom suivant :

**test\_apache\_playbook.yaml**

* + - 1. Ajoutez les informations suivantes au fichier. Assurez-vous que vous utilisez l'indentation YAML appropriée. Chaque espace et chaque tiret sont significatifs. Vous risquez de perdre une partie du formatage si vous faites un copier-coller.

**---**

- hosts: webservers

tasks:

- name: run echo command

command: /bin/echo hello world

### Exécutez le playbook Ansible pour tester votre groupe de webservers.

Exécutez le playbook Ansible à l'aide de la commande **ansible-playbook** à l'aide de l'option **-v** verbeuse. Vous devriez voir un résultat similaire à ce qui suit.

~/ansible/local$ **ansible-playbook -v test\_apache\_playbook.yaml**

Using /home/user01/ansible/local/ansible.cfg as config file

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.20.250]

TASK [run echo command] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250] => {"changed": true, "cmd": ["/bin/echo", "hello", "world"], "delta": "0:00:00.003718", "end": "2021-01-21 19:42:20.004095", "rc": 0, "start": "2021-01-21 19:42:20.000377", "stderr": "", "stderr\_lines": [], "stdout": "hello world", "stdout\_lines": ["hello world"]}

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

~/ansible/local$

### Créez votre playbook Ansible pour installer Apache.

* + - 1. Dans VS Code, créez un nouveau fichier dans le répertoire **local** avec le nom suivant :

**install\_apache\_playbook.yaml**

* + - 1. Ajoutez les informations suivantes au fichier. Assurez-vous que vous utilisez l'indentation YAML appropriée. Chaque espace et chaque tiret sont significatifs. Vous risquez de perdre une partie du formatage si vous faites un copier-coller. Le texte surligné est expliqué à l'étape suivante.

---

- hosts: webservers

become: yes

tasks:

- name: INSTALL APACHE2

apt: name=apache2 update\_cache=yes state=latest

- name: ENABLED MOD\_REWRITE

apache2\_module: name=rewrite state=present

notify:

- RESTART APACHE2

handlers:

- name: RESTART APACHE2

service: name=apache2 state=restarted

### Examinez votre playbook Ansible.

Voici une explication de certaines des lignes significatives de votre playbook:

* **hosts: webservers** - Ceci fait référence au groupe de périphériques de **webservers** dans votre fichier d'inventaire **hosts**. Ce playbook sera exécuté pour tous les appareils avec ce groupe.
* **become: yes** - Le mot-clé  **become** active l'exécution de la commande **sudo**, ce qui permet de réaliser des tâches telles que l'installation d'applications.
* **apt:** - Le module **apt** est utilisé pour gérer les paquets et les installations d'applications sous Linux.
* **handlers:** - Les gestionnaires sont similaires à une tâche, mais ne sont pas exécutés automatiquement. Ils sont appelés par une tâche. Notez que la tâche **ENABLED MOD\_REWRITE** appelle le gestionnaire **RESTART APACHE2**.

### Exécutez la sauvegarde Ansible pour installer Apache.

Exécutez le playbook Ansible à l'aide de la commande **ansible-playbook** à l'aide de l'option **-v** verbeuse. La première fois qu'Apache est installé sur votre machine virtuelle, la tâche **INSTALL APACHE2** prendra de 30 secondes à quelques minutes en fonction de votre vitesse Internet.

~/ansible/local$ **ansible-playbook -v install\_apache\_playbook.yaml**

**Question -** Est-ce que la commande a réussi ?

La commande ne fonctionne pas car on s’est branché sur … mais on a pas le mot de passe sudo. On a écrit le mot de passe SSH

Dans le fichier **hosts**, ajouter la variable ansible\_sudo\_pass=*VotreMotDePasse* (non, ce n’est pas la méthode la plus sécuritaire) à la suite des autres variables de votre hôte. Relancez la commande.

Vous auriez pu également utiliser la commande :

ansible-playbook -v install\_apache\_playbook.yaml --extra-vars "ansible\_sudo\_pass=*VotreMotDePasse*"

Le PLAY RECAP doit afficher **ok** et **failed=0** indiquant une exécution réussie du playbook.

~/ansible/local$ **ansible-playbook -v install\_apache\_playbook.yaml --extra-vars "ansible\_sudo\_pass=*VotreMotDePasse*"**

Using /home/user01/ansible/local/ansible.cfg as config file

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.20.250]

TASK [INSTALL APACH2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250] => {"cache\_update\_time": 1611277015, "cache\_updated": true, [sortie enlevée]

TASK [ENABLED MOD\_REWRITE] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250] => {"changed": true, "result": "Module rewrite enabled"}

[sortie enlevée]

RUNNING HANDLER [RESTART APACHE2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250] => {"changed": true, "name": "apache2", "state": "started",

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.20.250 : ok=4 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

Le PLAY RECAP doit afficher **ok** et **failed=0** indiquant une exécution réussie du playbook.

### Vérifiez qu’Apache a été installé.

* + - 1. Utilisez la commande suivante pour vérifier qu'Apache est maintenant installé. Appuyez sur "q" pour quitter.

~/ansible/local$ **sudo systemctl status apache2**

apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>

Active: active (running) since Wed 2020-05-20 03:48:49 UTC; 10min ago

Docs: https://httpd.apache.org/docs/2.4/

Process: 8201 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>

Main PID: 8225 (apache2)

Tasks: 55 (limit: 4654)

Memory: 5.3M

CGroup: /system.slice/apache2.service

├─8225 /usr/sbin/apache2 -k start

├─8229 /usr/sbin/apache2 -k start

└─8230 /usr/sbin/apache2 -k start

~/ansible/local$

* + - 1. Ouvrez un navigateur Web et entrez l'adresse IPv4 de votre nouveau serveur, **192.168.20.250**, pour afficher la page Web Apache2 par défaut.
      2. **Question - Inclure une capture d’écran** de la page par défaut d’Apache.

## Ajouter des options à votre Playbook Ansible pour webservers Apache

Dans un environnement de production, l'installation par défaut d'Apache2 est généralement personnalisée pour les fonctionnalités spécifiques requises par l'organisation. Un playbook Ansible peut également aider à automatiser ces tâches de configuration. Dans cette partie, vous allez personnaliser votre playbook en spécifiant que le serveur Apache utilise un numéro de port différent.

### Créez votre playbook Ansible pour installer Apache.

* + - 1. Dans VS Code, créez un nouveau fichier dans le répertoire **local** avec le nom suivant :

**install\_apache\_options\_playbook.yaml**

* + - 1. Ajoutez les informations suivantes au dossier. Assurez-vous que vous utilisez l'indentation YAML appropriée. Chaque espace et chaque tiret sont significatifs. Vous risquez de perdre une partie du formatage si vous faites un copier-coller.

---

- hosts: webservers

become: yes

tasks:

- name: INSTALL APACHE2

apt: name=apache2 update\_cache=yes state=latest

- name: ENABLED MOD\_REWRITE

apache2\_module: name=rewrite state=present

notify:

- RESTART APACHE2

- name: APACHE2 LISTEN ON PORT 8081

lineinfile:

path=/etc/apache2/ports.conf

regexp="^Listen 80"

line="Listen 8081"

state=present

notify:

- RESTART APACHE2

- name: APACHE2 VIRTUALHOST ON PORT 8081

lineinfile:

path=/etc/apache2/sites-available/000-default.conf

regexp="^<VirtualHost \\*:80>"

line="<VirtualHost \*:8081>"

state=present

notify:

- RESTART APACHE2

handlers:

- name: RESTART APACHE2

service: name=apache2 state=restarted

Cette liste de plays est très similaire à la précédente avec l'ajout de deux tâches que les serveurs Web écoutent sur le port 8081 au lieu du port 80.

Le module **lineinfile** est utilisé pour remplacer les lignes existantes dans les fichiers /etc/apache2/ports.conf et /etc/apache2/sites-available/000-default.conf.

**Question -** Faites une recherche dans la documentation Ansible pour le module **lineinfile**, décrire le module et chacun des paramètres utilisés.

lineinfile :

path :

regexp :

line :

state :

### Examinez les deux fichiers qui seront modifiés par le playbook.

Affichez les fichiers **/etc/apache2/ports.conf** et **/etc/apache2/sites-available/000-default.conf**. Notez que le serveur Web est actuellement à l'écoute sur le port 80.

~/ansible/local$ **cat -v /etc/apache2/ports.conf**

# If you just change the port or add more ports here, you will likely also

# have to change the VirtualHost statement in

# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl\_module>

Listen 443

<output omitted>

~/ansible/local$ **cat -v /etc/apache2/sites-available/000-default.conf**

<VirtualHost \*:80>

# The ServerName directive sets the request scheme, hostname and port that

# the server uses to identify itself. Ceci est utilisé lors de la création

# redirection URLs. Dans le contexte des hôtes virtuels, le nom du serveur

<output omitted>

~/ansible/local$

### Exécutez le Playbook Ansible.

* + - 1. Exécutez le playbook Ansible à l'aide de la commande **ansible-playbook**.

~/ansible/local$ **ansible-playbook install\_apache\_options\_playbook.yaml**

PLAY [webservers] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.20.250]

TASK [INSTALL APACHE2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.20.250]

TASK [ENABLED MOD\_REWRITE] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ok: [192.168.20.250]

TASK [APACHE LISTEN ON PORT 8081] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250]

TASK [APACHE2 VIRTUALHOST ON PORT 8081] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250]

RUNNING HANDLER [RESTART APACHE2] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

changed: [192.168.20.250]

PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

192.168.20.250 : ok=6 changed=3 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0

~/ansible/local$

### Vérifiez qu'Apache a été modifié.

* + - 1. Affichez à nouveau les fichiers **/etc/apache2/ports.conf** et **/etc/apache2/sites-available/000-default.conf**. Notez que le playbook a modifié ces fichiers pour les écouter sur le port 8081.

~/ansible/local$ **cat -v /etc/apache2/ports.conf**

# If you just change the port or add more ports here, you will likely also

# have to change the VirtualHost statement in

# /etc/apache2/sites-enabled/000-default.conf

Listen 8081

<IfModule ssl\_module>

Listen 443

</IfModule>

<IfModule mod\_gnutls.c>

Listen 443

</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

~/ansible/local$

~/ansible/local$ **cat -v /etc/apache2/sites-available/000-default.conf**

<VirtualHost \*:8081>

# The ServerName directive sets the request scheme, hostname and port that

# the server uses to identify itself. Ceci est utilisé lors de la création

# redirection URLs. Dans le contexte des hôtes virtuels, le nom du serveur

<output omitted>

~/ansible/local$

* + - 1. Ouvrez le navigateur Web et entrez l'adresse IPv4 de votre nouveau serveur. Mais cette fois, spécifiez 8081 comme numéro de port, **192.168.20.250:8081**, pour voir la page Web Apache2 par défaut.

**Remarque**: bien que vous puissiez voir dans le fichier **ports.conf** qu'Apache2 écoute également sur le port 443, il s'agit d'un protocole HTTP sécurisé. Vous n'avez pas configuré Apache2 pour un accès sécurisé.

* + - 1. **Question - Inclure une capture d’écran** de la page par défaut d’Apache au port 8081.

Fin du document

## Correction

|  |  |  |
| --- | --- | --- |
| **Item** | **Valeur (points)** | **Résultat (points)** |
| Partie 1 – Étape 1 | 2 |  |
| Partie 2 – Étape 1 | 3 |  |
| Partie 3 – Étape 5 | 4 |  |
| Partie 3 – Étape 6 | 4 |  |
| Partie 4 – Étape 1 | 3 |  |
| Partie 4 – Étape 4 | 4 |  |
| **Total** |  |  |

Compétences développées

|  |  |
| --- | --- |
| **00SK** – Effectuer le déploiement de serveurs Internet. | **00SJ # 1** – Analyser le projet de déploiement.  **00SJ # 2** – Monter les serveurs Internet.  **00SJ # 3** – Installer les services Internet.  **00SJ # 5** – Participer à la mise en service des serveurs Internet. |

**Note** : les compétences sont développées en partie.

Références

<https://docs.ansible.com/ansible/latest/index.html>