Mardi, le 10 Décembre 2024

DAMBE Lamboni

Option : Master 2 SSI

Groupe TP A

Module : Administration Réseaux et protocoles

dlamboni31@gmail.com

THÈME : Rapport du TP5, Ansible

NOTE	OBSERVATION

Étape 1 : Installation et configuration du conteneur via lxd

1. Création du conteneur LXD

lxc launch ubuntu:20.04 CodeIgniter4-Tp5
lxc exec CodeIgniter4-Tp5 -- bash

Le conteneur CodeIngniter4-Tp5 a été crée avec succès.

2. Configurattion de sshd_conf

Nous modifions la configration de ssh pour autoriser la connexion par mot de pas juste dans le cadre de ce tp mais nous savons très bien que la connexion pas clef est plus sécurisé.

sudo nano /etc/ssh/sshd_config

#Cette ligne a été décommenter et positionné à no par DAMBE Lamboni pour ne pas utiliser l'authentification par cle
PubkeyAuthentication no

#Ces deux lignes a été decommenter par DAMBE Lamboni pour activer la connection par mot de passe et en root
PasswordAuthentication yes
PermitRootLogin yes

#Cette ligne a été décommenter et positionner sur yes par DAMBE Lamboni pour permetre la resolution du chanllenge de mot de passe
ChallengeResponseAuthentication yes

Nous redémarrons le service ssh après ces modification en fin de les prendre en compte.

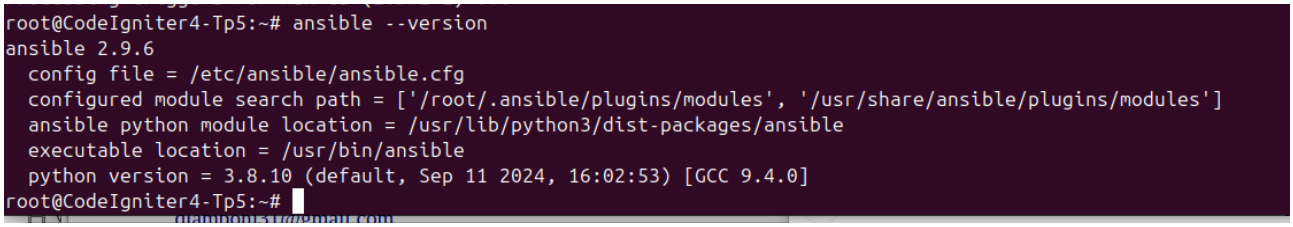
sudo systemctl restart sshd

Étape 2 : Installation des utilitaires sur le conteneur CodeIgniter4-Tp5

- **Ansible**

Nous installons ansible sur notre conteneur afin de pouvoir automatiser le déploiement de notre infrastructure via le play-book.

lxc exec CodeIgniter4-Tp5 -- bash
sudo apt update
sudo apt install ansible
ansible --version



Ansible version 2.9.6 a été bien installé sur notre système.

- **PyMysSQL**

Nous aurions besoin d'utiliser **PyMysSQL** de python 3, pour cela, nous procédons à son installation

lxc exec CodeIgniter4-Tp5 -- bash
apt update
apt install -y python3-pymysql

L'utilitaire a été bien installé.

- **SSHPASS**

sudo apt install sshpass

Nous arions également besoin de **sshpass** afin de pouvoir fournir le mot de passe en paramètre sans avoir besoin d'interaction avec l'utilisateur lors du déploiement de notre infrastructure.

Étape 3 : Automation du déploiement

1. Configuration de l'inventaire

Dans votre répertoire de travail Ansible, créez un fichier d'inventaire pour spécifier l'adresse IP du conteneur.

Nous avons crée dans notre repertoire de travail courant, l'inventaire dans lequel nous spécifions l'adresse IP.

nano inventory.ini

#Configuration défini par DAMBE Lamboni pour le TP5 Administration et Protocoles

[codeigniter_servers]

CodeIgniter4-Tp5 ansible_host=10.207.193.46 ansible_user=root ansible_ssh_pass=Admin/AinSiBle1? ansible_connection=ssh ansible_python_interpreter=/usr/bin/python3

```
 GNU nano 4.8                                  inventory.ini                                  Modified
#Configuration defini par DAMBE Lamboni pour le TP5 Admininistration et Protocoles
[codeigniter_servers]
CodeIgniter4-Tp5 ansible_host=10.207.193.46 ansible_user=root ansible_ssh_pass=Admin/AinSiBle1? ansible_connection=ssh ansible_python_int
```

2. Création et configurations des rôles ansible

Nous créons maintenant trois principaux rôles pour organiser les taches de notre infrastructure. Il s'agit notamment du rôle **webserver** pour le serveur web nginx, **php** pour PHP-FPM avec codeIgniter, et **mysql** pour la base de donné MySQL.

2.1 Création et configuration du rôle webserver

- **Création**

ansible-galaxy init webserver

```
root@CodeIgniter4-Tp5:~# ansible-galaxy init webserver
- Role webserver was created successfully
root@CodeIgniter4-Tp5:~#
```

Le rôle de notre serveur web a été bien crée.

- **Configuration**

Nous modifions le fichier **webserver/tasks/main.yml** pour installer et configurer notre serveur web nginx.

nano webserver/tasks/main.yml

```
---
- name: Install Nginx
  apt:
    name: nginx
    state: present
- name: Start and enable Nginx service
  systemd:
    name: nginx
    state: started
    enabled: yes
- name: Install PHP-FPM and necessary PHP modules
  apt:
```

```
    name:
      - php-fpm
      - php-mysql
      - php-cli
      - php-curl
    state: present
- name: Ensure PHP-FPM is running and enabled
  systemd:
    name: php7.4-fpm
    state: started
    enabled: yes
- name: Configure Nginx for CodeIgniter
  template:
    src: nginx_codeigniter.conf.j2
    dest: /etc/nginx/sites-available/codeigniter
  notify:
    - Reload Nginx
- name: Enable Nginx site configuration
  file:
    src: /etc/nginx/sites-available/codeigniter
    dest: /etc/nginx/sites-enabled/codeigniter
    state: link
  notify:
    - Reload Nginx
- name: Remove default site configuration
  file:
    path: /etc/nginx/sites-enabled/default
    state: absent
  notify:
```

- Reload Nginx

Synthèse :

Notre rôle configure un serveur web en automatisant l'installation et la configuration de nginx et PHP-FPM pour héberger notre application CodeIgniter. Il installe nginx, PHP-FPM, ainsi que les modules PHP requis (comme php-mysql et php-curl), tout en s'assurant que les services nécessaires sont démarrés et activés. Le rôle déploie une configuration personnalisée pour CodeIgniter à partir d'un modèle, active cette configuration, désactive le site par défaut de nginx, et applique les modifications en rechargant le service nginx via des handlers.

- **Création du template de configuration nginx**

Nous créons le fichier **nginx_codeigniter.conf.j2** dans le répertoire **webserver/templates/** avec une configuration de base pour Nginx et CodeIgniter que voici :

nano webserver/templates/nginx_codeigniter.conf.j2

```
server {
    listen 80;
    server_name {{ ansible_hostname }};
    root /var/www/html/codeigniter/public;
    index index.php index.html index.htm;
    location / {
        try_files $uri $uri/ /index.php?$query_string;
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.4-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
```

```
    location ~ /\.ht {
        deny all;
    }
}
```

Synthèse :

Notre template nginx configure un serveur virtuel pour héberger une application codeIgniter. Il écoute sur le port 80 avec le nom d'hôte dynamique défini par Ansible ({{ ansible_hostname }}) et utilise /var/www/html/codeigniter/public comme répertoire racine. Les fichiers index prioritaires sont index.php, index.html, et index.htm. Le routage dynamique est géré via try_files pour rediriger les requêtes vers index.php si nécessaire. Les fichiers PHP sont traités par PHP-FPM via un socket Unix, et une règle de sécurité bloque l'accès aux fichiers cachés comme .htaccess, renforçant la protection du serveur.

2.2 Création et configuration du rôle MySQL

- **Creation**

ansible-galaxy init mysql

Le rôle pour mysql a bien crée également

- **Configuration**

Nous modifions le fichier **mysql/tasks/main.yml** pour installer MySQL et créer la base de données et l'utilisateur.

nano mysql/tasks/main.yml

```
---
- name: Install MySQL
  apt:
    name: mysql-server
    state: present
- name: Ensure MySQL is running and enabled
  systemd:
    name: mysql
    state: started
```

```
    enabled: yes
- name: Create a MySQL database for CodeIgniter
  mysql_db:
    name: codeigniter_db
    state: present
- name: Create a MySQL user for CodeIgniter
  mysql_user:
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: "codeigniter_db.*:ALL"
    state: present
```

Synthèse :

Notre rôle MySQL automatise l'installation et la configuration du serveur MySQL pour prendre en charge une application CodeIgniter. Il installe le paquet mysql-server et s'assure que le service MySQL est démarré et activé. Ensuite, il crée une base de données nommée codeigniter_db et un utilisateur MySQL avec les identifiants définis par les variables mysql_user et mysql_password. Cet utilisateur reçoit tous les privilèges sur la base de données codeigniter_db, permettant à l'application de gérer ses données de manière sécurisée et efficace.

1.3 Création et configuration du rôle PHP

- **Création**

ansible-galaxy init php
Le rôle php est bien crée aussi pour notre codeIgniter 4

- **Configuration**

Nous éditons le fichier **php/tasks/main.yml** pour installer PHP et ses extensions nécessaires.

nano php/tasks/main.yml

```
---
- name: Install PHP 7.4 and required extensions
  apt:
    name:
      - php7.4
      - php7.4-cli
      - php7.4-mysql
      - php7.4-curl
    state: present
```

Synthèse :

Notre rôle PHP installe automatiquement PHP 7.4 et ses extensions nécessaires pour prendre en charge les applications codeIgniter. Il comprend les paquets essentiels comme `php7.4` pour le langage principal, `php7.4-cli` pour les outils en ligne de commande, `php7.4-mysql` pour la connexion à MySQL, et `php7.4-curl` pour gérer les requêtes HTTP. Ce rôle prépare un environnement PHP complet et optimisé pour le bon fonctionnement de l'application.

3. Création du playbook Principal

Notre playbook principal inclure les rôles que nous avons précédemment définis pour déployer l'application sur le serveur.

nano site.yml

```
---
- name: Deploy CodeIgniter 4 application with Nginx
  hosts: codeigniter_servers
  become: yes
  vars:
    mysql_password: "secret_password"
  roles:
```

- webserver

- php

- mysql

4. Déploiement

Nous déployons maintenant les fichiers de votre application codeIgniter dans notre conteneur . Ajoutez une tâche pour copier les fichiers dans le répertoire approprié.

- **webserver/tasks/deploy_codeigniter.yml**

nano webserver/tasks/deploy_codeigniter.yml

```
---
- name: Copy CodeIgniter 4 application to the server
  copy:
    src: ./AppCodeIgniter
    dest: /var/www/html/codeigniter
    owner: www-data
    group: www-data
    mode: '0755'
- name: Set proper permissions for CodeIgniter
  file:
    path: /var/www/html/codeigniter
    owner: www-data
    group: www-data
    recurse: yes
```

Synthèse :

notre playbook copie les fichiers de l'application CodeIgniter depuis un chemin local vers le répertoire `/var/www/html/codeigniter` sur le serveur. Il attribue les permissions nécessaires à l'utilisateur et au groupe `www-data` et définit les fichiers avec des permissions d'exécution (mode `0755`). Ensuite, il applique ces permissions de manière

récursive à tous les fichiers et sous-répertoires de l'application pour garantir que le serveur web puisse y accéder et les exécuter correctement.

- **webserver/handlers/main.yml**

nano webserver/handlers/main.yml

```
---
- name: Reload Nginx
  systemd:
    name: nginx
    state: restarted
```

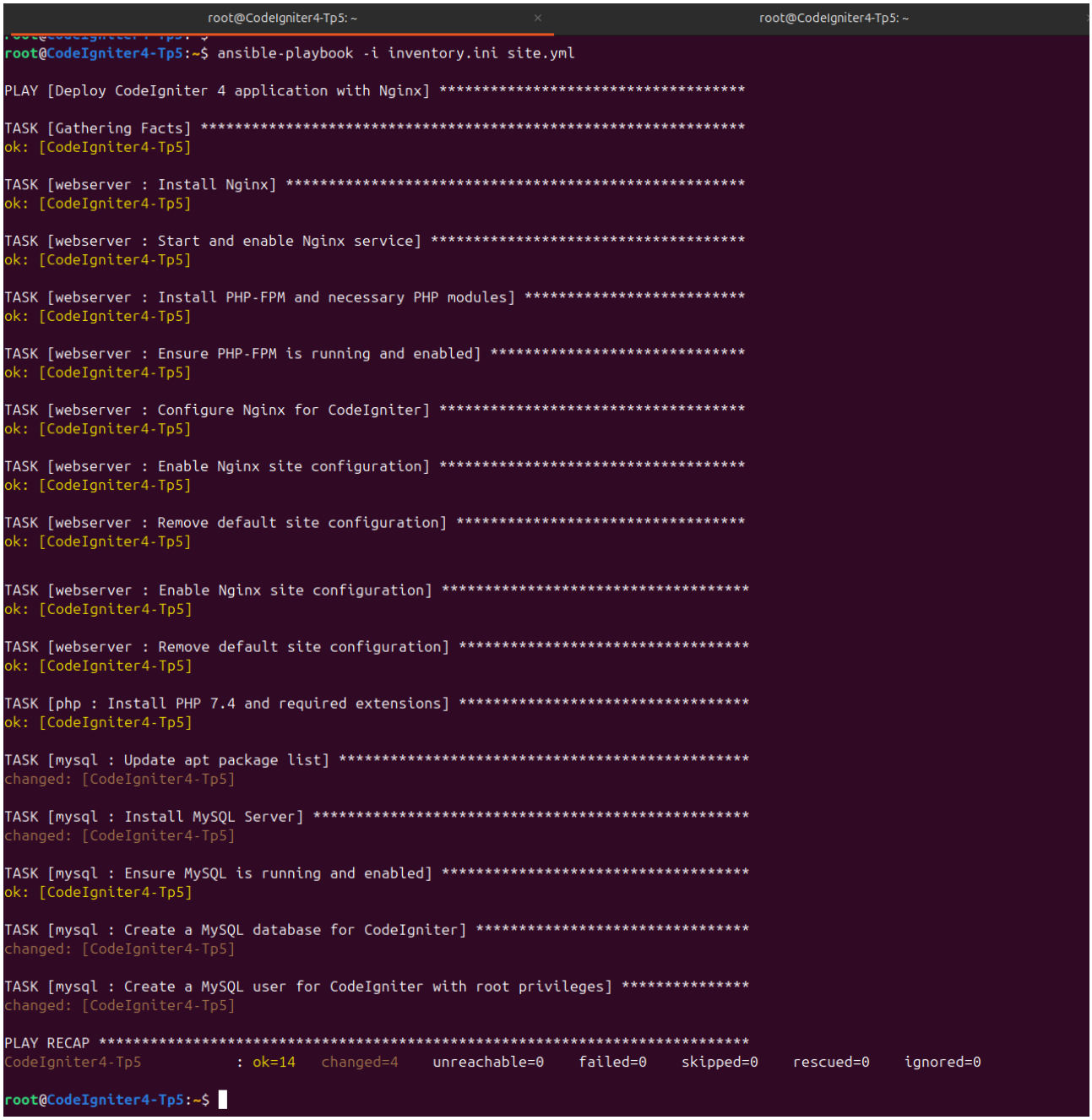
Synthèse:

Nous définissons la tâche qui redémarrera le service nginx en utilisant le module `systemd` avec l'état `restarted`. Son rôle est de garantir que nginx applique les modifications de configuration, telles que l'ajout ou la modification de sites, sans avoir besoin de redémarrer complètement la machine. En redémarrant proprement le service, elle assure que nginx prend en compte les dernières modifications de configuration et continue de fonctionner correctement avec les nouveaux paramètres.

5. Exécution de notre playbook

Nous déployons définitivement notre infrastructure en exécutant notre playbook.

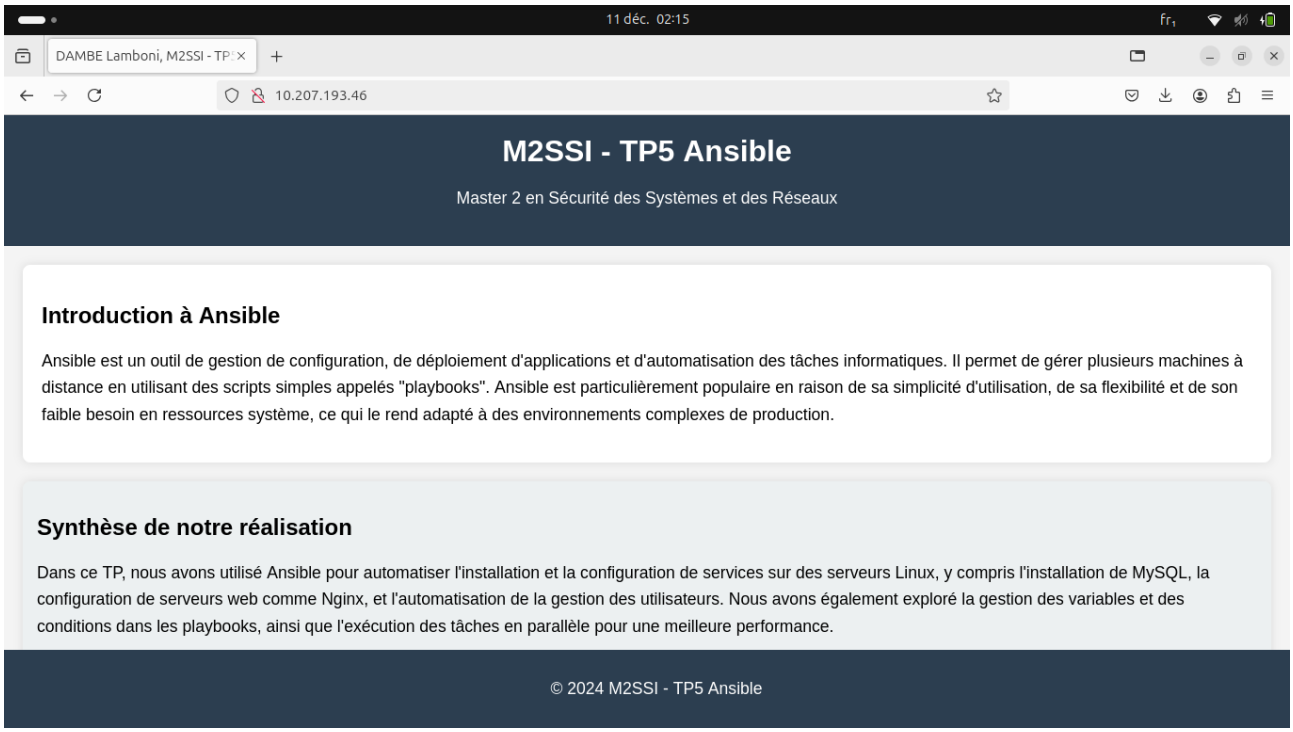
ansible-playbook -i inventory.ini site.yml


root@CodeIgniter4-Tp5: ~
root@CodeIgniter4-Tp5: ~
root@CodeIgniter4-Tp5:~$ ansible-playbook -i inventory.ini site.yml
PLAY [Deploy CodeIgniter 4 application with Nginx] ***********************************
TASK [Gathering Facts] ***
ok: [CodeIgniter4-Tp5]
TASK [webserver : Install Nginx] ***
ok: [CodeIgniter4-Tp5]
TASK [webserver : Start and enable Nginx service] ************************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Install PHP-FPM and necessary PHP modules] *************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Ensure PHP-FPM is running and enabled] *****************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Configure Nginx for CodeIgniter] ***********************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Enable Nginx site configuration] ***********************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Remove default site configuration] *********************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Enable Nginx site configuration] ***********************************
ok: [CodeIgniter4-Tp5]
TASK [webserver : Remove default site configuration] *********************************
ok: [CodeIgniter4-Tp5]
TASK [php : Install PHP 7.4 and required extensions] *********************************
ok: [CodeIgniter4-Tp5]
TASK [mysql : Update apt package list] ***
changed: [CodeIgniter4-Tp5]
TASK [mysql : Install MySQL Server] **
changed: [CodeIgniter4-Tp5]
TASK [mysql : Ensure MySQL is running and enabled] ***********************************
ok: [CodeIgniter4-Tp5]
TASK [mysql : Create a MySQL database for CodeIgniter] *******************************
changed: [CodeIgniter4-Tp5]
TASK [mysql : Create a MySQL user for CodeIgniter with root privileges] **************
changed: [CodeIgniter4-Tp5]
PLAY RECAP ***
CodeIgniter4-Tp5 : ok=14 changed=4 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
root@CodeIgniter4-Tp5:~$

Vérification du bon fonctionnement

```
root@CodeIgniter4-Tp5:~$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.40-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show schemas;
+--------------------+
| Database           |
+--------------------+
| codeigniter_db     |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.05 sec)

mysql>
```

Nous voyons bien que notre base de donné CodeIgniter été bien crée et notre application s'exécute correctement comme le montre cette capture requêtant sur notre site.



Références :

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

https://openclassrooms.com/fr/courses/2035796-utilisez-ansible-pour-automatiser-vos-taches-de-configuration/6373897-assemblez-les-operations-avec-les-playbooks-pour-automatiser-le-deploiement

https://iac.goffinet.org/ansible-linux/un-premier-playbook/