

Projet Ansible : Déploiement d'un site web avec Nginx

Objectif

Créer un playbook Ansible capable de :

- Installer et configurer un serveur web Nginx
- Déployer un site web HTML simple
- S'assurer que le service reste actif
- Utiliser des variables et des templates Jinja2 pour automatiser le contenu

Infrastructure

- Deux machines :
 - **Serveur** : où Ansible est installé
 - **Target** : machine cible qui recevra la configuration
- Les deux machines sont sur le même réseau **host-only** (192.168.159.0)

```
xavier@serveur:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/xavier/.ssh/id_ed25519):
Enter passphrase for "/home/xavier/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/xavier/.ssh/id_ed25519
Your public key has been saved in /home/xavier/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:04b9G8zXXmMliqDrbko9zqvvaEtGPBpPiikQI/fpTo xavier@serveur
The key's randomart image is:
+--[ED25519 256]--+
|
|      .          +
|    . o .  S +   . .
|  . o = . . + = . o .
|.. =.o+. . . * .oo|
|+ oE=o=B .   +..o|
|+. o=%@o+   .   .|
+-----[SHA256]-----+
```

Création d'une clé SSH sur le serveur

```
xavier@serveur:~$ ssh-copy-id xavier@192.168.159.129
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/xavier/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
xavier@192.168.159.129's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'xavier@192.168.159.129'"
and check to make sure that only the key(s) you wanted were added.
```

Copie de la clé publique sur la target pour permettre une connexion **sans mot de passe**

```
xavier@serveur:~$ ssh xavier@192.168.159.129
Linux target 6.12.57+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.57-1 (2025-11-05) x86_64

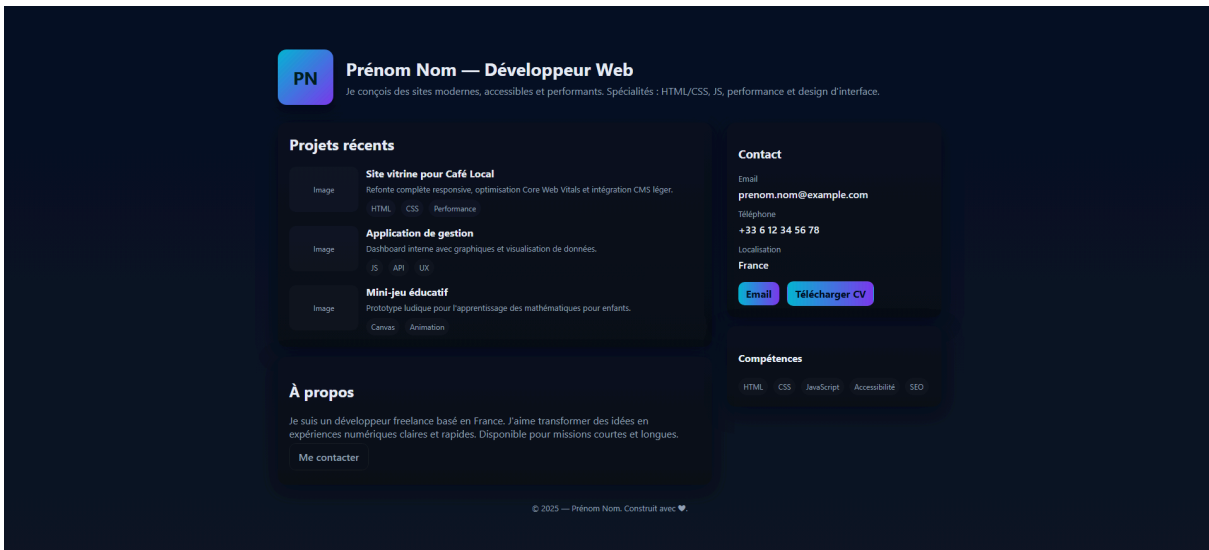
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Nov 28 09:00:07 2025 from 192.168.159.128
xavier@target:~$ |
```

Test réussi : le serveur peut se connecter à la target sans entrer de mot de passe

```
> index.html
! inventory.yml
⚠ playbook.yml
```

pour commencer le index.html va etre le site que je veut déployé



Le site HTML à déployer est préparé (généré via ChatGPT pour simplifier)

```
webservers:
  hosts: 192.168.159.129
  vars:
    ansible_user: xavier
    ansible_become: yes
    ansible_become_password: [REDACTED]
```

Fichier d’inventaire défini : Spécifie l’hôte cible, l’utilisateur et le mot de passe initial pour l’installation

```

- name: Install and configure Nginx
  hosts: webservers
  tasks:
    - name : installation nginx
      apt :
        name : nginx
        state : present
    - name: Ensure Nginx is running
      service:
        name: nginx
        state: started
        enabled: yes

    - name: Deploy custom homepage
      copy:
        src: index.html
        dest: /var/www/html/index.html
        notify: Restart nginx
  handlers:
    - name: Restart nginx
      service:
        name: nginx
        state: restarted

```

Hôte ciblé : webserver défini dans l'inventaire

Tâches :

Installer Nginx (state: present)

1. Vérifier que le service est actif (state: started)
2. Déployer le site HTML : copier le fichier index.html dans /var/www/html/index.html
3. Appeler un **handler** pour redémarrer Nginx et appliquer les changements

Une fois tout précisé, il suffit de lancer le playbook avec :

```

xavier@serveur:~/nginx-projet$ ansible-playbook playbook.yml -i inventory.yml

```

```
xavier@serveur:~/nginx-projet$ ansible-playbook playbook.yml -i inventory.yml

PLAY [Install and configure Nginx] *****

TASK [Gathering Facts] *****
[WARNING]: Host '192.168.159.129' is using the discovered Python interpreter at '/usr/bin/python3.13',
but future installation of another Python interpreter could cause a different interpreter to be disco
vered. See https://docs.ansible.com/ansible-core/2.19/reference_appendices/interpreter_discovery.html
for more information.
ok: [192.168.159.129]

TASK [installation nginx] *****
changed: [192.168.159.129]

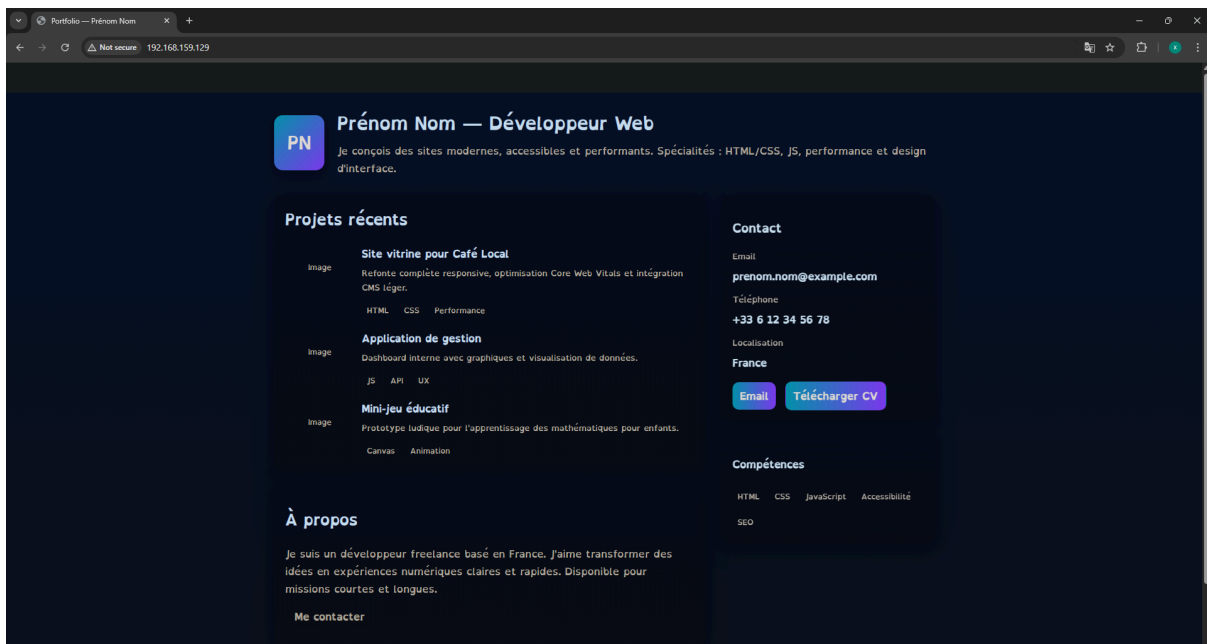
TASK [Ensure Nginx is running] *****
ok: [192.168.159.129]

TASK [Deploy custom homepage] *****
changed: [192.168.159.129]

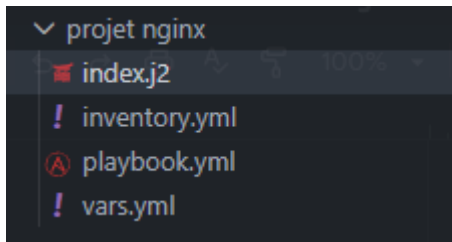
RUNNING HANDLER [Restart nginx] *****
changed: [192.168.159.129]

PLAY RECAP *****
192.168.159.129 : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
```

Chaque étape est visible lors de l'exécution, ce qui permet de suivre le déroulé et de vérifier que tout fonctionne



En accédant à l'IP de la target dans un navigateur, le site apparaît correctement



Pour rendre le projet plus dynamique, le fichier HTML a été remplacé par un template **Jinja2** (.j2)

```
1  owner_name: "Xavier Neveu"
2  métier : "étudiant"
3  owner_initials: "XN"
4  lead_text: "Je teste les variables Ansible"
5  about_text: "J'aime bien Ansible"
6  email: "xavier.neveu@example.com"
7  phone: "+33 6 12 34 56 78"
8  location: "France"
9  skills:
10     - HTML
11     - CSS
12     - JavaScript
13     - Accessibilité
14     - SEO
15  projects:
16     - title: "Déploiement de site avec Ansible"
17       description: "Déploiement d'un site basique, puis en mettant des variables"
18       tags: ["HTML", "Ansible", "Nginx"]
19
```

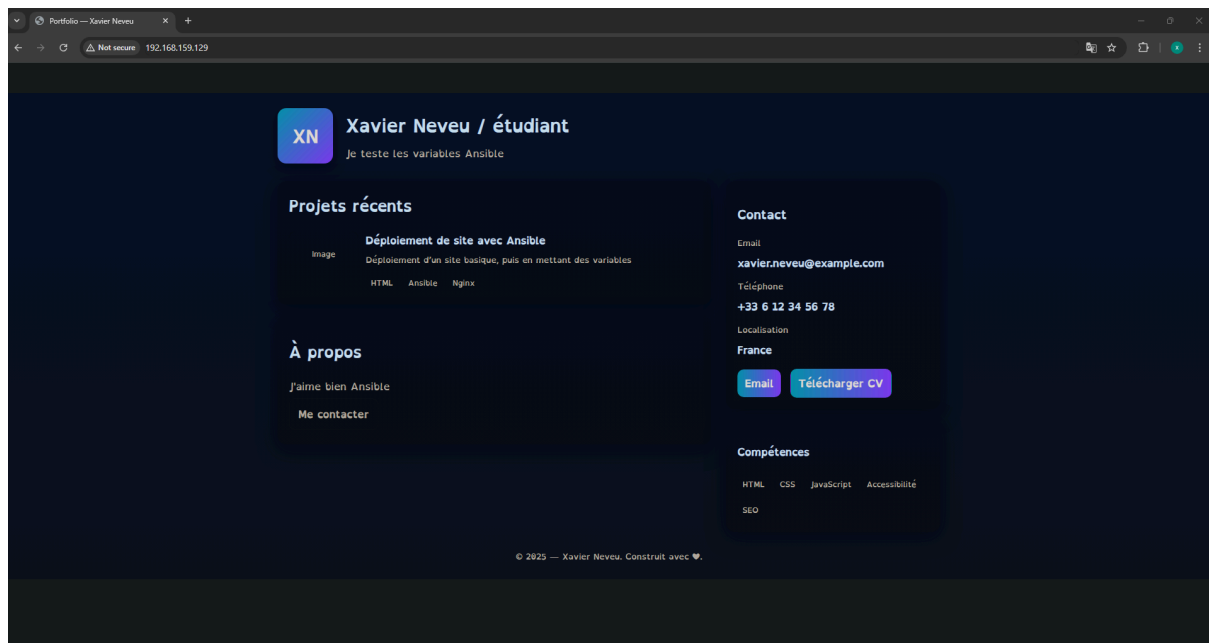
Les variables sont définies dans un fichier `vars.yml`

```
- name: Deploy custom homepage
  template:
    src: index.j2
    dest: /var/www/html/index.html
  notify: Restart nginx
```

Le playbook a été modifié pour utiliser **template:** au lieu de **copy:**

```
xavier@serveur:~/nginx-projet$ ansible-playbook playbook.yml -i inventory.yml -e @vars.yml
```

Le playbook est lancé en précisant le fichier de variables avec `-e @vars.yml` (ou directement via la ligne de commande pour quelques variables)



Résultat : le site est automatiquement généré avec les valeurs définies dans le fichier de variables

Résumé

- Infrastructure prête avec SSH sans mot de passe
- Playbook Ansible configuré pour installer Nginx et déployer le site
- Variables et templates utilisés pour rendre le site dynamique et facilement modifiable
- Déploiement simple et automatisé en lançant une seule commande